Optimierung der Datenvorverarbeitung in neuronalen Netzwerken zur Bildqualitätsbewertung

Masterarbeit

Studiengang Systemtechnik

der Hochschule Ruhr West

Fabian Karrer

10007660

Erstprüfer:	Prof. Dr.-Ing. Zhichun Lei
Zweitprüfer:	M.Sc. Thomas Thuilot

Vorgelegt am 26. Dezember 2023

Duisburg, Dezember 2023

Zusammenfassung

Im Rahmen dieser Arbeit wird zunächst ein Überblick über die Grundlagen des maschinellen Lernens und über verschiedene Methoden der digitalen Bildqualitätsbewertung gegeben.

Eine dieser Methoden wird näher betrachtet und es werden mögliche Probleme für das Training von neuronalen Netzwerken, insbesondere von Netzwerken zur Bildqualitätsbewertung herausgearbeitet.

Dieses Problem wird quantifiziert und es werden zwei grundlegende Verfahren zur Lösung entworfen. Dabei wird jeweils ein separates Konzept entwickelt und ausgewertet.

Die beiden Ansätze werden in einem direkten Vergleich unter verschiedenen Gesichtspunkten gegenübergestellt und es wird der vielversprechendere davon ausgewählt.

Der ausgewählte Lösungsansatz wird implementiert und in ein bereits existierendes Netzwerk zur Bildqualitätsbewertung integriert. Die Leistung des modifizierten Netzwerkes wird mit dem Original verglichen.

Abstract

This thesis first provides an overview of the basics of machine learning and various methods of digital image quality assessment.

One of these methods is examined in more detail and possible problems for the training of neural networks, in particular networks for image quality assessment, are identified.

This problem is quantified and two basic methods for solving it are designed. In each case, a separate concept is developed and evaluated.

The two approaches are compared directly from different points of view and the more promising one is selected.

The selected approach will be implemented and integrated into an existing image quality assessment network. The performance of the modified network is compared with the original.

Inhaltsverzeichnis

Zusammenfassung II
Abstract III
Inhaltsverzeichnis IV
Abbildungsverzeichnis V
Tabellenverzeichnis VI
1 Einleitung 1
1.1 Motivation 1
1.2 Problemstellung und Zielsetzung 2
1.3 Gliederung und Vorgehensweise 3
2 Stand der Technik 4
2.1 Machine Learning und neuronale Netzwerke 4
2.2 Image Quality Assessment 8
2.2.1 TReS 11
3 Methodik zur Verbesserung der Datenvorverarbeitung 14
3.1 Bewertung 14
3.2 Random Cropping und Formulierung des Problems 16
3.3 Ansatz 1: Entropievergleich 19
3.3.1 Methoden- und Konzeptentwicklung 19
3.3.2 Auswertung des Entropievergleichs 25
3.4 Ansatz 2: Feature Matching 27
3.4.1 Methoden- und Konzeptentwicklung 28
3.4.2 Auswertung des Feature Matchings 39
3.5 Vergleich der Ansätze 42
4 Implementation 45
4.1 Grundlagenoptimierung 45
4.2 Entropievergleich 46
4.3 Feature Matching 49
5 Auswertung des Gesamtsystems 51
6 Fazit 53
Literaturverzeichnis 54
Erklärung 57

Abbildungsverzeichnis

Abbildung 1 Einfaches neuronales Netzwerk [2] 6
Abbildung 2: Donald Michies Maschine zum Erlernen des Spiels Tic Tac Toe [4] 7
Abbildung 3 Aufbau der Feature-Verarbeitung durch den Transformer [22] 12
Abbildung 4 Architektur von TReS [25] 12
Abbildung 5 Beispiel der grafischen Bewertungsoberfläche 15
Abbildung 6 Beispiele verschiedener Random Crops 16
Abbildung 7 Random Crop ohne Motiv des Bildes 17
Abbildung 8 Beispiel für den Informationsgehalt eines Bildes 19
Abbildung 9: Python-Code zur Berechnung der Entropie eines 1D-Signals 20
Abbildung 10 Entropiebeispiel mit Random Crops 20
Abbildung 11 Auszüge der Testbilder 21
Abbildung 12 Verteilung der Entropiedifferenz für positive Treffer 22
Abbildung 13 Verteilung der Entropiedifferenz für negative Treffer 22
Abbildung 14 Anteil der negativen Matches nach Entropiedifferenz 23
Abbildung 15 Verteilung der Entropiedifferenz 24
Abbildung 16 Feature Matching eines Flugzeugs 27
Abbildung 17 Fehlgeschlagenes Feature Matching eines Cocktailglases 28
Abbildung 18 Besseres Feature Macthing des Cocktailglases 29
Abbildung 19 Wüste ohne erkennbare Merkmale 30
Abbildung 20 Mond zur Tageszeit ohne erkennbare Merkmale 31
Abbildung 21 Bildausschnitt beinhaltet keine Merkmale 31
Abbildung 22 Bildausschnitt Helikopter 32
Abbildung 23 Feature Matching Fehlerkategorien 33
Abbildung 24 Zusammensetzung der Fehler des Typ 2 34
Abbildung 25 Histogramm der positiven Übereinstimmungen 35
Abbildung 26 Histogramm der negativen Übereinstimmungen 36
Abbildung 27 Verteilung für einen Feature-Matching Grenzwert von 0% 37
Abbildung 28 Verteilung für einen Feature-Matching Grenzwert von <2% 37
Abbildung 29 Geschwindigkeitsvergleich beim Feature Matching 39
Abbildung 30 Geschwindigkeitsvergleich von Entropie und Feature Matching 42
Abbildung 31 Vergleich der Genauigkeit 43
Abbildung 32 Ablaufdiagramm pil_loader() 46
Abbildung 33 Ablaufdiagramm der Entropieberechnung 47
Abbildung 34 Ablaufdigramm des Entropievergleichs 48
Abbildung 35 Ablaufdiagramm des Feature Matchings 50
Abbildung 36 Training von TReS ohne Modifikationen 51
Abbildung 37 Training von TReS mit Entropiedifferenz-Filter 52

Tabellenverzeichnis

Tabelle 1 Genauigkeit von TReS nach [21] 13
Tabelle 2 Aufbau verschiedener IQA_Datensätze [18] 14
Tabelle 3 Gegenüberstellung der Verhältnis-Grenzwerte 38
Tabelle 4 Vergleich der beiden Ansätze 44

1 Einleitung

Künstliche Intelligenz (KI) nimmt in unserem alltäglichen Leben eine zunehmend größer werdende Rolle ein. Dabei ist die Verarbeitung von und das Arbeiten mit Bildern eines der populärsten Einsatzgebiete. Hier siedelt sich das Feld des Image Quality Assessment (kurz IQA) an.

Beim Image Quality Assessment handelt es sich um die Bewertung der Qualität eines Bildes. IQA findet in fast allen Teilgebieten der modernen Bildverarbeitung in verschiedenen Formen Anwendung. Eines der häufigsten Einsatzgebiete ist die Verwendung als Rückmeldung zum Einstellen von Parametern und Optimieren von Algorithmen in Systemen, wie Kameras in Mobiltelefonen oder anderen Bildquellen. Aber auch in Bereichen wie dem autonomen Fahren wird IQA immer wichtiger.

Diese Arbeit beschäftigt sich mit der Optimierung der Eingangsdaten, mit denen solche Systeme trainiert werden.

1.1 Motivation

Der Stellenwert der Bildqualitätsbewertung im gesamten Feld der Bildverarbeitung hat in den letzten Jahren zusehends zugenommen. Dies ist unter anderem dem erhöhten Einsatz von künstlicher Intelligenz in Geräten, wie Mobiltelefonen, aber auch in anderen Bild-basierten Produkten, zu verdanken.

Bei näherer Betrachtung mehrerer Netzwerke, im Rahmen einer anderen Arbeit, fällt jedoch auf, dass es bei weitgehend einfarbigen Bildern, oder Bildern mit kleinen Motiven und weitgehend eintönigen Hintergründen, häufiger zu Fehlbewertungen kommt als bei inhaltlich diversen Bildern. Um diesem Problem auf den Grund zu gehen, wurden diese einzelnen Netzwerke näher untersucht.

Dabei fällt auf, dass bei allen betrachteten Netzwerken, wie allgemein üblich, Verfahren zur Diversifizierung der Eingangsdaten eingesetzt werden. Diese können jedoch potenziell die entsprechenden Eingangsdaten verfälschen. Insbesondere das gängige Random Cropping, also die zufällige Wahl eines Ausschnittes des Bildes, wird als mögliche Fehlerquelle eingeschätzt.

Das Problem wird im nachfolgenden Kapitel näher erläutert.

Dieses potenzielle Problem näher zu betrachten und mögliche Lösungen oder Verbesserungen zu finden ist die Motivation dieser Arbeit.

1.2 Problemstellung und Zielsetzung

Random Cropping bezeichnet die Praxis, statt dem ganzen Bild, einen, oder auch mehrere, Ausschnitte eines Bildes zum Trainieren des Netzwerks zu nutzen.

Dieses Verfahren erhöht zum einen die Größe des Datensatzes, da mehr verschiedenen Bilder verwendet werden, zum anderen wird so häufig auch die Robustheit des Netzwerkes verbessert, da Fälle wie verschiedene Kamerapositionen oder teilweise Verdeckungen so emuliert und in den Eingangsdaten widergespiegelt werden.

Bei anderen gängigen Anwendungsfällen, wie der Klassifizierung von Bildern, ist dieses Verfahren mit verhältnismäßig wenigen Risiken verbunden. Die Motive sind hier im Regelfall klar erkennbar und die Zuordnung erfolgt in vorgegebene Kategorien.

Bei der Bildqualitätsbewertung lassen sich die Bilder jedoch nicht klaren Gruppen zuordnen, sondern der Inhalt des Bildes beeinflusst direkt den Bildqualitätswert. So ist das Risiko deutlich höher, dass der Qualitätswert des Originalbildes nicht mehr repräsentativ für den Bildausschnitt ist und das Netzwerk mit fehlerhaften Daten trainiert wird.

Dieses Problem ist nicht spezifisch auf die Bildqualitätsbewertung, sondern es kann davon ausgegangen werden, dass das Training mit fehlerhaften Daten allgemein ein Problem in mehreren Anwendungsbereichen ist.

Im Rahmen der vorliegenden Arbeit soll zunächst das Problem quantifiziert werden.

Anschließend soll ein Verfahren erarbeitet werden, das es ermöglicht diese abweichenden Eingangsdaten beim Training zu erkennen und herauszufiltern. Eine anteilmäßige Reduzierung dieser Fälle ist das primäre Ziel dieser Arbeit. Die Lösung sollte dabei möglichst einfach und universell anwendbar gehalten werden, um einen Einsatz bei anderen Anwendungsfällen zu erleichtern.

Ein weiteres Ziel der Arbeit ist der Einsatz der erarbeiteten Lösung in einem gängigen Netzwerk zur Bildqualitätsbewertung und der Vergleich mit bereits existierenden Ergebnissen. Da diese Ergebnisse aber stark variabel sind, und aufgrund der zeitlich begrenzten Dauer der Arbeit nur bedingt Versuche möglich sind, werden konkrete Verbesserungen der Genauigkeit nur als sekundäres Ziel betrachtet.

1.3 Gliederung und Vorgehensweise

Die Arbeit wird, beginnend mit der Einleitung, in insgesamt sechs Kapitel unterteilt.

In Kapitel 2.1 wird zunächst ein Überblick über die Grundlagen des Themas Machine Learning und im Spezielleren neuronale Netzwerke gegeben, um eine Basis zum Verständnis der restlichen Arbeit zu schaffen. Es wird auch kurz die historische Entwicklung des Gebietes erläutert.

Da der Fokus dieser Arbeit im Bereich des Image Quality Assessments (IQA) liegt, wird in Kapitel 2.2 näher auf diesen eingegangen. Dabei werden anhand einer groben historischen Entwicklung verschiedene Verfahren und Ansätze vorgestellt.

Kapitel 2.2.1 gibt einen tieferen Einblick in das IQA-Netzwerk TReS, welches im weiteren Verlauf der Arbeit von Bedeutung ist.

Die Quantifizierung des Problems und die Auslegung möglicher Problemlösungen erfolgt in Kapitel 3. Dazu werden in Kapitel 3.1 zunächst die Notwendigkeit und die Voraussetzungen für ein objektives Bewertungskriterium herausgestellt und dieses festgelegt.

In Kapitel 3.2 wird das Verfahren „Random Cropping“ tiefer beleuchtet und die hierdurch entstehenden möglichen Probleme werden quantifiziert.

Anschließend werden mehrere Konzepte zur Problemlösung entworfen und individuell ausgewertet.

Kapitel 4 thematisiert die Implementierung der in Kapitel 3 erarbeiteten Konzepte. Dabei liegt der Fokus auf dem Aufbau der Lösung und der Integration in bereits existierende Netzwerke. Die Darstellung erfolgt mittels Erklärungen der Programmabläufe, Auszügen aus dem Quellcode und entsprechenden Diagrammen.

Der so entstandene Prototyp wird nun ausgewertet. Dazu wird das Netzwerk in Kapitel 5 mit mehreren Datensätzen trainiert und die Leistung mit den nicht modifizierten Versionen des Netzwerks verglichen.

Abschließend wird ein Fazit gezogen und es werden mögliche Anregungen, Verbesserungen und Komplikationen erläutert.

2 Stand der Technik

In diesem Kapitel werden zunächst die grundlegenden Funktionenweisen und die historische Entwicklung von Machine Learning und neuronalen Netzen erläutert. Anschließend wird näher auf die verschiedenen Möglichkeiten zur Qualitätsbewertung durch diese eingegangen, wobei vor allem auf NR (No-Reference) -Methoden eingegangen wird. Zuletzt wird exemplarisch auf das konkrete Netzwerk TReS (Transformers, Relative Ranking, and Self-Consistency) eingegangen, auf dem viele der in dieser Arbeit vorgeschlagenen Methoden basieren und getestet werden.

2.1 Machine Learning und neuronale Netzwerke

Machine Learning

Machine Learning ist ein Sammelbegriff für die Befähigung von Computern, Probleme zu lösen, ohne explizit darauf programmiert zu werden. Der Computer „entwirft“ seinen eigenen Algorithmus, um zu einer funktionierenden Lösung zu kommen.

Ein üblicher Machine Learning Algorithmus besteht grob aus drei Teilen [1].

1. Einem Entscheidungsprozess: Es muss ein Algorithmus, oder eine Folge von Aktionen, vorhanden sein, der aus gegebenen Eingangsdaten ein Ergebnis erhält.
2. Eine Fehlerfunktion: Die Fehlerfunktion bewertet die Vorhersage des Entscheidungsprozesses. Wenn Beispiele bekannt sind, können diese mit der Vorhersage verglichen werden, um den Grad der Abweichung einzuschätzen.
3. Ein Optimierungsprozess: Der Optimierungsprozess nutzt das Ergebnis der Fehlerfunktion, um den Entscheidungsprozess so anzupassen, dass die Abweichung des nächsten Ergebnisses geringer ist.

Durch ein vielfaches Wiederholen dieser drei Schritte wird die Genauigkeit des Modells, zumindest theoretisch, immer weiter gesteigert.

Insgesamt unterscheidet man im Allgemeinen zwischen vier verschiedenen Ansätzen des Lernens [1]:

- Supervised machine learning

  Die zur Verfügung gestellten Daten wurden vor Beginn des Trainings, meistens durch, oder unter Aufsicht von Menschen, mit “Labeln” versehen, die den Inhalt der Daten genau definieren. Das Modell versucht seinen Output möglichst genau auf diese Label einzustellen. Beispiele für diese Art des Machine Learnings sind die Klassifizierung von Bildern, oder akustische Spracherkennung.

- Unsupervised machine learning

  Wie der Name bereits vermuten lässt, beinhalten die Trainingsdaten beim Unsupervised Learning keine Label. Diese Art des Machine Learnings wird genutzt, um ohne menschliche Hilfe versteckte Muster und Strukturen in Daten zu finden. Praktische Beispiele sind der Einsatz in der explorativen Datenanalyse, oder die Erkennung von Gesichtern in Bildern.

- Semi-supervised machine learning

  Beim Semi-supervised Learning handelt es sich um eine Kombination aus den beiden vorherigen Kategorien. Es gibt einen kleineren, gelabelten Datensatz, der und einen großen Datensatz ohne Label. Diese Art des Lernens hat ähnliche Einsatzgebiete, wie das Supervised Learning, benötigt jedoch deutlich weniger eindeutig gelabelte Daten. Dies hilft, wenn es entweder nicht möglich ist, ausreichend Label zu bestimmen, oder das individuelle Labeln aller Daten zu aufwendig wäre.

- Reinforcement learning

  Beim Reinforcement Learning werden dem Algorithmus keine Trainingsdaten bereitgestellt. Stattdessen muss der Algorithmus während seines Einsatzes aus seinen Fehlern und Erfolgen lernen. So ist das Modell gezwungen selbst herauszufinden welche Folge von Aktionen zu einem positiven Ergebnis führt. Der Vorteil dieses Ansatzes ist, dass der Algorithmus selbstständig und in einer unbekannten Umgebung in der Lage ist ein Problem zu lösen. Anwendungsfälle hierfür sind beispielsweise Roboter, oder die Empfehlungen von Plattformen, wie YouTube oder Netflix.

Eine weitere Möglichkeit ist das sogenannte Deep Learning. Dieses basiert auf sogenannten neuronalen Netzwerken, die im nachfolgenden Absatz erläutert werden und beschreibt im Speziellen Netzwerke, die mehrere Hidden Layers beinhalten.

Hierbei handelt es sich um ein neueres Feld, bei dem automatisch und ohne grundlegende Regeln, oder menschliches Wissen, aus Datensätzen gelernt wird. Dieser Ansatz benötigt jedoch sehr große Mengen an Rohdaten und die Genauigkeit steigt im Regelfall mit der Menge der zur Verfügung stehenden Daten.

Der häufigste Anwendungsfall für diese Art des Machine Learnings ist bei Problemen, die für Menschen eher intuitiv lösbar sind und sich nicht durch mathematische Regeln beschreiben lassen. Beispiele dafür sind Bilderkennung, Spracherkennung, oder Sprachverständnis.

Neuronale Netzwerke

Neuronale Netzwerke sind eine Unterkategorie des Machine Learnings, die sich, gerade in letzter Zeit, gegenüber anderen Ansätzen hervorgehoben hat.

Dabei handelt es sich um, dem menschlichen Gehirn nachempfundene, Vernetzung künstlicher Neuronen. Es geht dabei aber weniger um die Nachbildung der biologischen Struktur als um die Emulation der Informationsverarbeitung des Gehirns.

Verbindungen von biologischen Neuronen werden über die Gewichtungen zwischen den Knotenpunkten der künstlichen Neuronen nachgebildet (siehe Abbildung 1). Der Zusammenschluss solcher Neuronen bildet ein System, das Computer dazu befähigt aus ihren Fehlern zu lernen und die Lösung für das bestimmte Problem kontinuierlich zu verbessern.

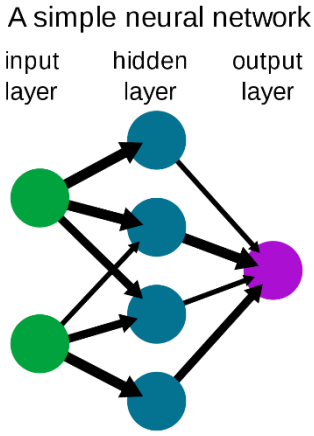


Abbildung 1 Einfaches neuronales Netzwerk [2]

Auch wenn es sich hierbei um ein sehr modernes Forschungsgebiet handelt, ist das grundlegende Konzept keinesfalls neu. Nahezu zeitgleich mit der Einführung programmierbarer Computer entstand bereit die Idee, das menschliche Gehirn elektronisch nachzubauen. Das erste mathematische Modell eines Neurons wurde 1943 von Warren McCulloch und Walter Pitts vorgestellt [3]. Damit war es möglich grundlegende logische Verknüpfungen, wie AND, OR oder NOT umzusetzen.

In den 1950ern und 1960ern erfuhr das Feld der künstlichen Intelligenz großes öffentliches und wissenschaftliches Interesse. So stellte beispielsweise Donald Michie 1963 seine Maschine “Matchbox Educable Noughts And Crosses Engine”, kurz “MENACE” vor, die, zunächst nur aus Streichholzschachteln bestehend, in der Lage war, von einer zufälligen Startposition ausgehend das Spiel Tic Tac Toe zu erlernen. (siehe Abbildung 2)

Dazu wurden mögliche Spielzüge mit Gewichtungen versehen, die angepasst wurden, wenn die Maschine ein Spiel verlor oder gewann.

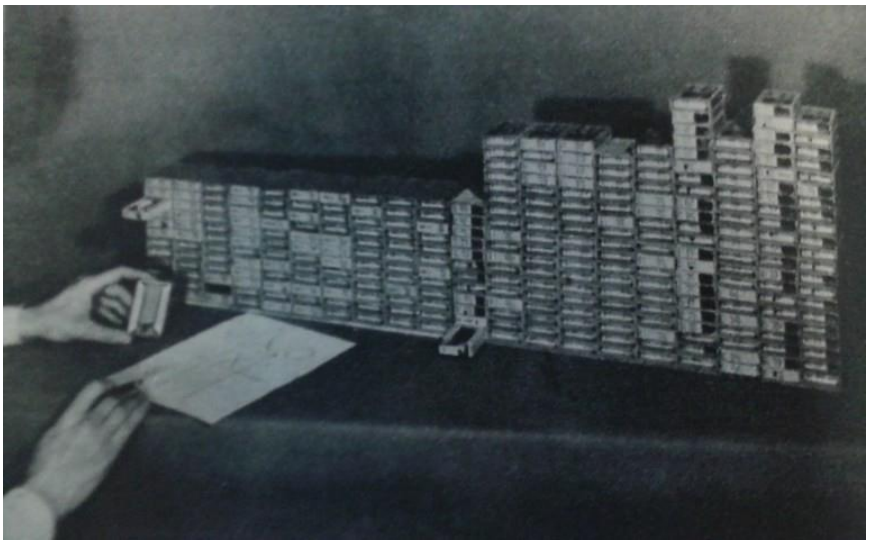


Abbildung 2: Donald Michies Maschine zum Erlernen des Spiels Tic Tac Toe [4]

Entwicklungen wie diese setzen hohe Erwartungen, denen das Feld aber nicht gerecht werden konnte. Grund dafür waren hauptsächlich die Limitierungen durch nicht ausreichende Rechenleistung der damals verfügbaren Computer.

Dies führte Anfang der 1970er Jahre zusehends zu Frustration unter Investoren und Organisationen und zu einem Sinken der Fördermittel, was wiederum zu sinkenden Forschungsbemühungen führte. Die 1970er werden heute als “erster AI-Winter” bezeichnet. Diese Zeit der Frustration hielt bis etwa Anfang der 1980er an.

Anfang der 1980er führten jedoch neue Entwicklungen und Ansätze zu erneutem, wenn auch eher zaghaftem, Optimismus. Eine maßgebliche Entwicklung, die neues Leben in das Feld brachte, war der praktische Einsatz des Backpropagation Algorithmus, der den Fehler des Netzwerks abhängig von den verschiedenen Gewichtungen durch das Netz zurückführen konnte.

Eine weitere Neuheit war der Einsatz von sogenannten „Hidden Layers“, die es den Netzwerken ermöglichten, deutlich komplexere Probleme zu lösen. Hidden Layers ist eine Bezeichnung für eine Lage Neuronen, die weder direkt am Eingang noch am Ausgang liegt und somit von außen nicht direkt „sichtbar“ ist. So konnten Ende der 1980er die ersten Schritte zum sogenannten “Deep Learning” gemacht werden.

Deep Learning bezeichnet eine Unterkategorie von Netzwerken, die, in ihrer Funktionsweise einem rudimentären Gehirn nachempfunden, viele Layer besitzen und mit großen Mengen an Daten trainiert werden. Ein Meilenstein, den diese Neuerungen ermöglichten, war in den 90ern die Entwicklung des ersten CNN, das in der Lage war, handgeschriebene Zahlen zu erkennen. [5]

In den folgenden Jahren entwickelte sich das Feld immer weiter, wobei der Fokus häufig auf individuellen Kategorien, wie Support-Vector-Maschinen [6] oder neuronalen Netzwerken, lag.

Die zunehmende verfügbare Rechenleistung, insbesondere von Grafikkarten, die besonders bei paralleler Matrixmultiplikation zum Tragen kommt, und die steigende Einfachheit, Netzwerke mit solchen Grafikkarten zu trainieren beschleunigte die Entwicklung zusätzlich.

Heutzutage stellt Machine Learning eine der wichtigsten technologischen Entwicklungen der modernen Zeit dar. Sie findet in diversen Industrie- und Forschungsbereichen, vom Bankenwesen [7] bis zur Krebsforschung [8], Anwendung. Auch neue aufkommende Technologien, wie autonome Fahrzeuge [9] oder die zunehmend besser werdenden Möglichkeiten, menschliches Verhalten nachzubilden [10], wären ohne den Einsatz von maschinellem Lernen nicht möglich.

2.2 Image Quality Assessment

Die Qualität eines Bildes ist ein wichtiger Faktor in einer ganzen Reihe an Disziplinen der digitalen Bildverarbeitung. Darunter fällt, unter anderem, bereits das bloße Aufnehmen, oder auch das Komprimieren von Bildern. [11]

Aber auch spezifischere Anwendungen, wie die Verarbeitung von Bildern in autonomen Fahrzeugen [12], wo qualitativ minderwertige Daten potenziell zu Fehlinterpretationen der Umgebung und gefährlichen Situationen für den Fahrer und das Fahrzeug führen können, sind Fälle, in denen eine robuste Bewertung der Bildqualität essenziell ist.

Das Ziel beim Image Quality Assessment ist es immer eine Bewertung der Qualität von Bildern zu ermöglichen. Dazu existiert eine große Anzahl an Methoden, die sich in subjektive und objektive Bewertung aufteilen.

Subjektive Bewertungen basieren auf der Zuordnung durch einen Menschen und sind daher zeitaufwendig und erlauben keine vollständige Automatisierung des Systems. [13] Sie liefern zwar die höchste Genauigkeit, aber da die Systeme für die meisten Anwendungsfälle in Echtzeit laufen müssen, ist diese Methode oft ungeeignet und es werden in den meisten Fällen objektive Methoden eingesetzt.

Objektiv bedeutet dabei, dass möglichst keine maschinellen Tendenzen und Vorurteile vorhanden sind, sondern der Algorithmus selbstständig so nah an der subjektiven Bewertung durch einen durchschnittlichen Menschen arbeitet, wie möglich. Der Inhalt des Bildes und die Art möglicher Verzerrungen sollen dabei unerheblich sein.

Die objektive Bildbewertung unterteilt sich in drei Gruppen, je nachdem, ob ein Referenzbild oder andere Informationen für die Bewertung vorhanden sind, oder nicht.

Die drei Gruppen sind:

- Full-Reference (FR)

  Full-Reference-Methoden benötigen ein Referenzbild, bei dem davon ausgegangen wird, dass dieses in perfekter Qualität vorliegt. Das zu prüfende Bild wird dann mit diesem Referenzbild verglichen.

- No-Reference (NR)

  Bei No-Reference-Methoden ist es nicht notwendig, ein Referenzbild bereitzustellen. Die Bewertung durch das System erfolgt sozusagen blind.

- Reduced-Reference (RR)

  Auch bei Reduced-Reference-Methoden ist kein Referenzbild nötig, jedoch werden von dem System zusätzliche Informationen benötigt. Diese können beispielsweise extrahierte Merkmale sein, die dem System helfen die Qualität zu bestimmen. [13]

Da die Arbeit sich aber ausschließlich mit No-Reference-Bildbewertung befasst, wird nur darauf in mehr Detail eingegangen.

No-Reference-IQA lässt sich abermals grob in zwei Gruppen unterteilen.

Eine davon sind Ansätze, die sich auf spezifische Arten der Qualitätsminderung fokussieren. Bei diesen Qualitätsminderungen handelt es meistens sich um verschiedene Arten der Verzerrung, wie Verschwommenheit, oder Bildrauschen.

Die Einsatzfähigkeit in lebensnahen Anwendungen ist jedoch eher begrenzt, da sich die Art der Verzerrung bei realen Bedingungen meist eher schlecht voraussagen lässt.

Die andere Gruppe sind universelle Methoden. Diese werden, unabhängig von der Art der Verzerrung, anhand von extrahierten Features auf zuvor von Menschen erstellte Mean-Opinion-Scores (MOS) trainiert und sind so in der Lage mit unbekannten, oder gemischten Arten der Bildqualitätsverminderung zurecht zu kommen.

Ein Überblick über viele, wenn auch inzwischen eher veraltete, Methoden wird in [14] gegeben. Die meisten davon lassen sich allerdings eher der ersten Gruppe zuordnen, und sind nur in der Lage bestimmte Verzerrungstypen zu bewerten.

Auch gibt es große Unterschiede zwischen den verschiedenen Datensätzen, die zum Training genutzt werden. Diese lassen sich generell zwei Kategorien zuordnen. Synthetische und authentische Datensätze.

Synthetische enthalten Verzerrungen und Modifikationen, wie Rauschen, oder Unschärfe, die nachträglich auf Bild angewendet wurden. Beispiele für solche Datensätze sind [15] [16] [17]. Authentische Datensätze hingegen verwenden Bilder, die nicht gezielt verzerrt wurden, sondern „in-the-wild“, also nicht extra zur Erstellung des Datensatzes, aufgenommen wurden. Beispiele hierfür sind [18] [19] [20].

Authentische Datensätze sind zwar aufwendiger zu erstellen, ermöglichen es aber Netzwerke für realistischere Anwendungsfälle zu trainieren. Sie gelten aufgrund ihrer Objektivität und Realitätsnähe als Goldstandard.

Algorithmen, die tatsächlich universell anwendbar sind und gute Ergebnisse erzielen können, wenn sie auf authentische Datensätze angewendet werden, wurden erst in den letzten Jahren durch die Implementation von Deep Learning wirklich möglich.

Zwei Beispiele für solche Algorithmen sind:

- HyperIQA [21]:

  HyperIQA versucht den menschlichen Prozess der Qualitätsbewertung zu emulieren, indem es die eigentliche Bewertung von dem Verständnis des Bildinhaltes trennt. Somit erreicht HyperIQA eine der besten Genauigkeiten in authentischen Datensätzen.

- TReS [22]:

  Bei TReS (kurz für *T*ransformers, *Re*lative ranking and *S*elf consistency) handelt es sich um ein Netzwerk, welches CNNs und Transformer kombiniert um eine der besten, wenn nicht die beste, Genauigkeit bei authentischen Datensätzen zu erreichen. TReS wurde als exemplarisches Netzwerk für die im Rahmen dieser Arbeit erstellten Konzepte verwendet. Auf die genauere Struktur, Funktionsweise und Leistung wird im nächsten Teilkapitel näher eingegangen.

2.2.1 TReS

Bei TReS handelt es sich um ein referenzloses Netzwerk zur Bildqualitätsbestimmung (NR-IQA), welches 2020 von Golestaneh et al. veröffentlich wurde [22]. TReS steht dabei für *T*ransformers, *Re*lative Ranking und *S*elf consistency. Diese Punkte stellen die drei grundlegenden Methoden des Netzwerks dar, die es von anderen Ansätzen unterscheidet.

Transformer

TReS nutzt Transformer, um gängige Schwächen von rein CNN-basierten Lösungen zu kompensieren. CNNs sind dafür bekannt, eher lokale, als globale Strukturen und Merkmale in Bildern zu erfassen. Da IQA aber ein Fall ist, in dem sowohl lokale als auch globale Merkmale von großer Bedeutung sind, versucht TReS so, diesem Problem entgegenzuwirken. [22]

Relative Ranking

Eine weitere Eigenschaft von TReS ist der Einsatz von sogenanntem Relative Ranking. Dabei werden die Bilder nicht nur individuell betrachtet, sondern auch im Kontext der anderen Bilder eingeordnet.

Self consistency

Self-Consistency beschreibt das Vorgehen, mehrere mögliche und verschiedene Pfade eines Modells zu generieren und die konsistenteste Antwort zu nutzen. Mit dieser Technik lässt sich die Zuverlässigkeit und Robustheit des Netzwerks erhöhen.

Trainingsablauf

TReS extrahiert zunächst die Merkmale eines Bildes mittels eines CNN (Convolutional Neural Network). Dazu wird ResNet [23] als Backbone genutzt. Die Merkmale werden jeweils von der letzten Ebene der jeweiligen Blöcke gewonnen.

Da sie so aber verschiedene Skalierungen und Dimensionen haben, werden sie anschließend normalisiert, die Dimensionen werden vereinheitlicht und es wird Dropout angewendet, um möglichem Overfitting entgegenzuwirken.

Beim Dropout werden während des Trainings eine bestimmte Anzahl an Neuronen pro Layer deaktiviert und bei der Berechnung nicht berücksichtigt.

Als nächstes werden diese Merkmale zusammengeknüpft und an einen Transformer weitergegeben. Der Aufbau dieses Transformers orientiert sich an [24].

Da Transformer aber nicht auf die Ordnung der Eingangsdaten achten, werden zusätzlich Positionsdaten verwendet. Diese ermöglichen es dem Netzwerk auch, auf Informationen zur Position der wichtigsten Merkmale zurückzugreifen.

Der Aufbau des Transformers ist in Abbildung 3 dargestellt.

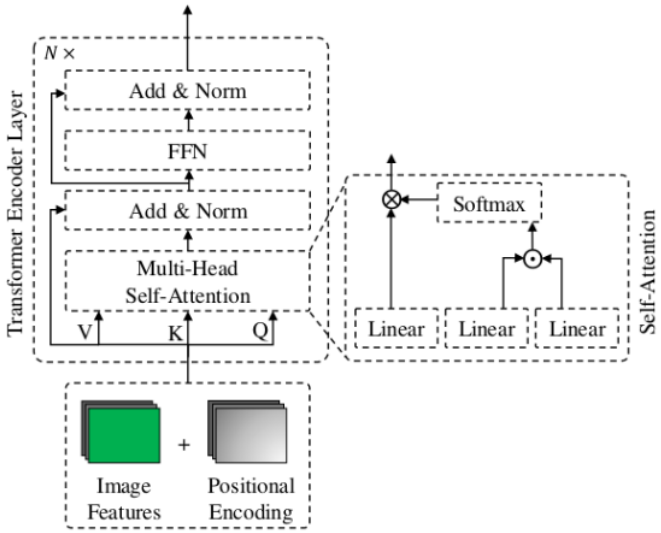


Abbildung 3 Aufbau der Feature-Verarbeitung durch den Transformer [22]

Um die durch das CNN gewonnen lokalen Merkmale und die durch den Transformer globalen Merkmale gemeinsam zu verarbeiten, werden sie mittels einer Fully-Connected (FC) Ebene zusammengefasst und es wird die Qualität vorhergesagt.

Der Aufbau des gesamten Netzwerks ist in Abbildung 4 gezeigt.

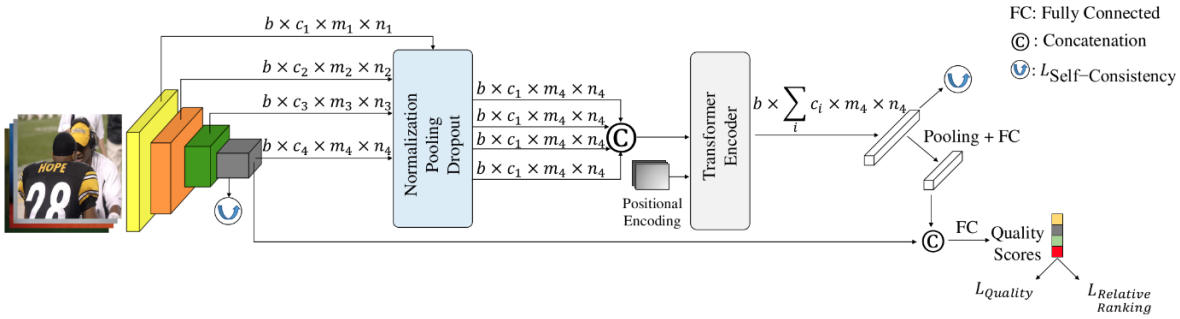


Abbildung 4 Architektur von TReS [25]

Das Training und die Evaluation von TReS erfolgt anhand von sieben, öffentlich verfügbaren, Datensätzen. Darunter sind sowohl synthetische als auch authentische Datensätze. Als synthetische Datensätze werden LIVE [11], CSIQ [17], TID2013 [16] und KADID-10K [26] genutzt. Die authentischen Datensätze sind CLIVE [27], KonIQ-10K [19] und LIVE-FB [18].

Die Implementation von TReS erfolgt in Python und stützt sich hauptsächlich auf das Paket PyTorch. Beim Training werden von jedem Bild im Trainingsdatensatz 50 zufällige Ausschnitte mit einer Größe von $224x224$ Pixeln gewählt. Jeder Ausschnitt erbt die Qualitätsbewertung des Originalbildes.

Von den Bildern des Datensatzes werden 80% zum Training und 20% zum Verifizieren der Ergebnisse zugeteilt. Die Zuteilung erfolgt zufällig vor Beginn des Trainings.

Genauigkeit

Die Ergebnisse des Netzwerks bestehen aus den Mittellungen von 10 Trainingsdurchläufen mit verschiedenen Seeds, die zur Festlegung der Trainings- und Testdaten genutzt werden.

Mit diesen Parametern ist TReS laut [22] fähig die in Tabelle 1 gezeigten Genauigkeiten zu erreichen. Die Genauigkeit wird in PLCC (Pearson correlation coefficient) und SROCC (Spearman correlation coefficient) angegeben.

Tabelle 1 Genauigkeit von TReS nach [21]

Datensatz	PLCC	SROCC
LIVE	0,968	0,969
CSIQ	0,942	0,922
TID2013	0,883	0,863
KADID	0,858	0,859
CLIVE	0,877	0,846
KonIQ	0,928	0,915
LIVEFB	0,625	0,554
Weighted Average	0,732	0,685

Mit diesen Ergebnissen erreicht TReS im eigenen Vergleich im Durchschnitt die höchste Genauigkeit unter den verglichenen Netzwerken.

Es ist anzumerken, dass diese Genauigkeit im eigenen Test nicht ganz erreicht werden konnte. Insbesondere die erreichten SROCC-Werte lagen konsistent unterhalb der Ergebnisse des Papers. Da die Differenz aber nicht groß war, wird davon ausgegangen, dass es sich dabei um normale Varianz handelt.

3 Methodik zur Verbesserung der Datenvorverarbeitung

3.1 Bewertung

Wie bereits in Kapitel 2.2 erläutert, ist das Ziel der referenzlosen Bildqualitätsbewertung (NR-IQA), insbesondere von Systemen, die universell einsetzbar sein sollen, die Emulation einer Bewertung der Bildqualität durch einen Menschen. Um dieses Ziel zu erreichen, stützten sich die Netzwerke im Regelfall auf sogenannte MOS-Scores. Dabei handelt es sich um die Bewertung einzelner Bilder durch mehrere Menschen, deren Bewertung gemittelt wird.

So lässt sich aus der subjektiven Einschätzung mehrerer Menschen eine möglichst objektive Bewertung bilden. Insbesondere bei Regressionsproblemen, zu denen auch die Bildqualitätsbewertung gehört, ist ein solcher Ansatz häufig die einzige Option.

Dadurch, dass aber jede Einschätzung der Qualität auf diesem Verfahren beruht, ist es wichtig, dass jegliche anderen Einschätzungen der Qualität, insbesondere bei direkten Vergleichen, oder Änderungen am Netzwerk, mit einem möglichst ähnlichen Verfahren durchgeführt werden. So lässt sich eine möglichst hohe Vergleichbarkeit der Ergebnisse sicherstellen.

Tabelle 2 zeigt den Aufbau mehrerer häufig eingesetzter IQA Datensätze.

Tabelle 2 Aufbau verschiedener IQA_Datensätze [18]

Datenbank	Art der Verzerrung	Anzahl der Bilder	Anzahl verzerrter Bilder	Bewertungen pro Bild	Art der Bewertung
CSIQ	künstlich	30	866	5-7	Labor
TID2013	künstlich	25	3000	9	Labor
KADID-10k	künstlich	81	10125	30	Crowds.
LIVE	authentisch	29	779	23	Labor
KonIQ	authentisch	10073	10073	120	Crowds.

Die Gewinnung der Bewertungen des Datensatzes KonIQ [19] wird exemplarisch näher betrachtet, da er sowohl aus authentischen Daten besteht und die größte Diversität der verglichenen Datensätze aufweist.

KonIQ besteht aus 10073 individuellen Bildern, die nicht zusätzlich verzerrt wurden. Der Qualitätswert für jedes Bild setzt sich aus 120 individuellen Bewertungen zusammen, die durch Probanden gewonnen und gemittelt wurden.

Die Bilder sind der YFCC100M [28] Datenbank entnommen. Um die individuellen Bewertungen zu bekommen, wurden insgesamt 2302 Probanden genutzt. Davon wurden 843 durch mehrere Teststufen als unzuverlässig herausgefiltert, sodass die Bewertungen durch 1459 Einzelpersonen gewonnen wurden.

So erreicht KonIQ eine hohe Diversität und Zuverlässigkeit seiner Bewertungen.

In Ermangelung von Zeit und Ressourcen kann im Rahmen dieser Arbeit keine vergleichbare Zuverlässigkeit erreicht werden. Eine Bewertung durch mehrere Probanden und anschließende Mittelung der Ergebnisse wird trotzdem als notwendig erachtet.

Auch der Aufbau der grafischen Oberfläche ist für die Einschätzung der Probanden wichtig. Um den Bildausschnitt möglichst frei vom Bildkontext darzustellen, wird darauf verzichtet eine bloße Auswahl im Originalbild abzubilden, sondern es werden das Original und der Ausschnitt als einzelne Bilder nebeneinander dargestellt.

Abbildung 5 zeigt die Anordnung der Bilder.

Abbildung 5 Beispiel der grafischen Bewertungsoberfläche

Die Probanden werden mittels Tastatureingabe dazu angehalten die Bilder, beziehungsweise den Bildausschnitt, unter dem jeweiligen Kriterium zu bewerten. Die Einschätzung erfolgt dabei binär in den Kategorien TRUE und FALSE. Zusätzlich gibt es noch die Kategorie MAYBE, um den Probanden eine Möglichkeit zu geben, bei schwer zu bewertenden Ausschnitten, ihre Unsicherheit zum Ausdruck zu bringen.

Die Anzahl Probanden pro Test variiert dabei, es werden jedoch im Mittel 5 verschiedenen Einschätzungen einbezogen.

3.2 Random Cropping und Formulierung des Problems

Das Erstellen mehrerer zufälliger Ausschnitte aus einem Bild ist eine gängige Methode zur Diversifizierung von Daten. Es ist damit möglich seinen Datensatz, um ein Vielfaches zu vergrößern, ohne eine deutlich größere Menge an originalen Bildern, oder Rechenleistung aufzuwenden. Abbildung 6 zeigt Beispiele für mögliche Random Crops in einem Bild.

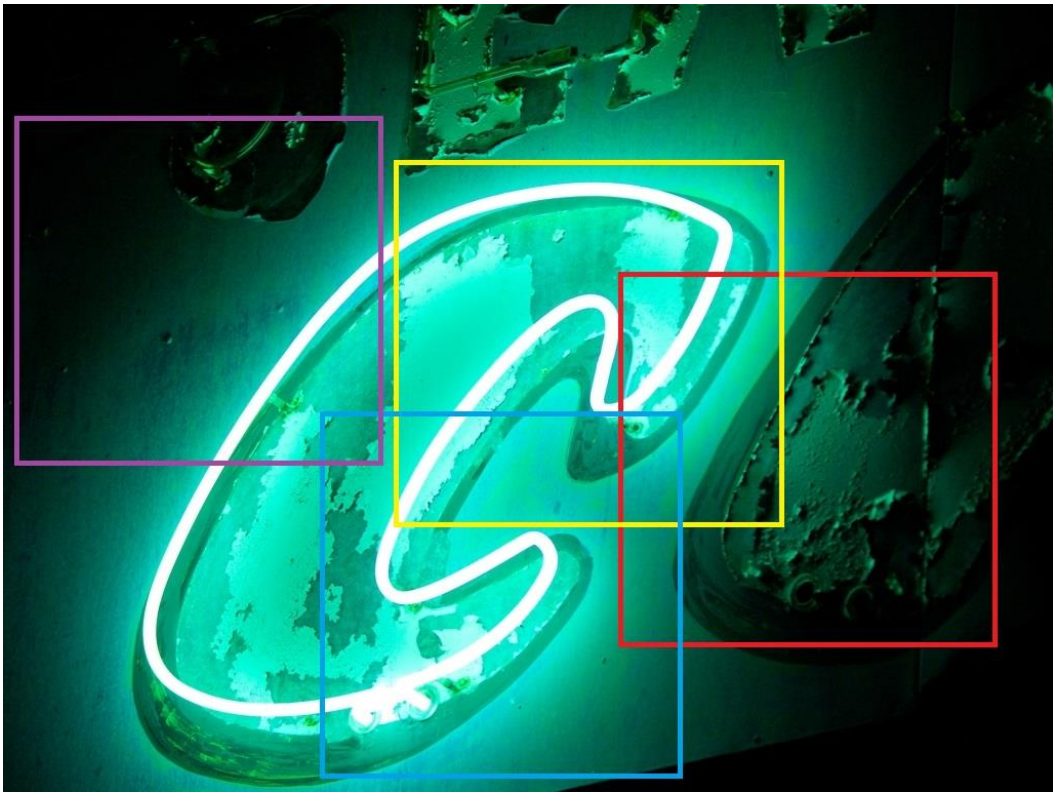


Abbildung 6 Beispiele verschiedener Random Crops

Studien zeigen, dass die Leistung von Netzwerken logarithmisch mit der Größe der Datensätze ansteigt [29]. Auch wird so die Chance des sogenannten “Overfitting” vermindert, da das Netzwerk gezwungen ist mehr zu generalisieren.

Jedoch birgt dieses Verfahren auch Risiken. So kann es passieren, dass dabei eine Region gewählt wird, die nicht den eigentlichen Inhalt des Bildes beinhaltet oder repräsentiert und somit die Daten verfälscht werden.

Ein extremes Beispiel für solch einen Fall ist in Abbildung 7 dargestellt.

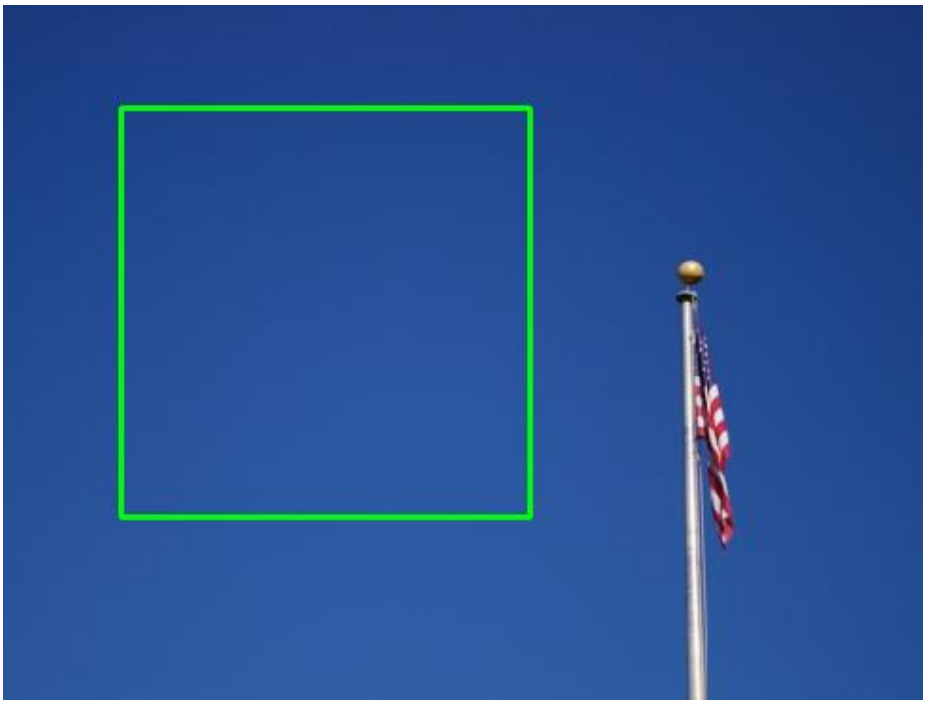


Abbildung 7 Random Crop ohne Motiv des Bildes

Insbesondere beim Image Quality Assessment, wo die Qualitätsmerkmale nicht unbedingt gleichförmig über das ganze Bild verteilt sind, kann es somit passieren, dass das Netzwerk mit fehlerhaften Daten trainiert wird.

Die meisten Hochleistungs-Netzwerke, wie TreS [22] oder MANIQA [30], nutzen random crops um die Daten zu diversifizieren und auf die notwendige Größe (bei ResNet-Backbone 224x224 Pixel) zu bringen.

Hierbei kommt es jedoch, vor allem bei bestimmten Bildtypen des Öfteren dazu, dass dieses problem auftritt und der gewählte Ausschnitt nicht das eigentliche Motiv des Bildes widerspiegelt.

Um herauszufinden, auf welchen Anteil der Ausschnitte dies zutrifft, werden 1047 zufällig gewählte Ausschnitte durch mehrere Probanden bewertet. Das Kriterium dabei ist, ob der Ausschnitt den gleichen Qualitätswert erhalten sollte wie das Originalbild. Das Originalbild und der Ausschnitt werden dabei, wie in Kapitel 3.1 erläutert, nebeneinander gezeigt.

Zur Bewertung wird eine einfache Mehrheit der individuellen Einschätzungen genutzt.

66 der Ausschnitte werden mehrheitlich so eingeschätzt, dass sie die Qualität des Originals nicht widerspiegeln. So ergibt sich ein Anteil von $6{,}3\%$ an Ausschnitten, die nicht qualitativ repräsentativ für das Gesamtbild sind.

Bei der Betrachtung dieser Ausschnitte fällt außerdem auf, dass diese häufig auftreten, wenn das Hauptmotiv nur einen kleinen Teil des Bildes einnimmt, oder große Teile des Bildes von einfarbigen Flächen eingenommen werden. Solche Bildtypen sind häufig Bilder mit fliegenden Hauptmotiven vor einem blauen Himmel, Szenen bei Nacht oder beispielsweise bei Schnee.

Zur Verbesserung des Problems kann ein Vergleich zwischen dem zufälligen Ausschnitt und dem Gesamtbild angestellt werden.

Hier gibt es mehrere mögliche Methoden zur Bestimmung der Ähnlichkeit zwischen dem originalen Bild und dem Teilausschnitt. Die wichtigsten Kriterien sind hierbei vor allem die Genauigkeit und Robustheit des Ansatzes, aber auch der notwendige Rechenaufwand, da der Vergleich dynamisch beim Training des Netzwerkes laufen muss.

Diese Möglichkeiten werden in dieser Arbeit vorgestellt, ausgearbeitet und verglichen.

3.3 Ansatz 1: Entropievergleich

Eine mögliche Methode zur Bestimmung der Ähnlichkeit zwischen Gesamt- und Teilbild ist der Vergleich des Informationsgehaltes beider Bilder.

Abbildung 8 zeigt ein Beispiel für die Informationsverteilung in einem Bild. Blau bedeutet dabei eine niedrige Entropie um den entsprechenden Pixel und rot eine hohe Entropie in der Umgebung.

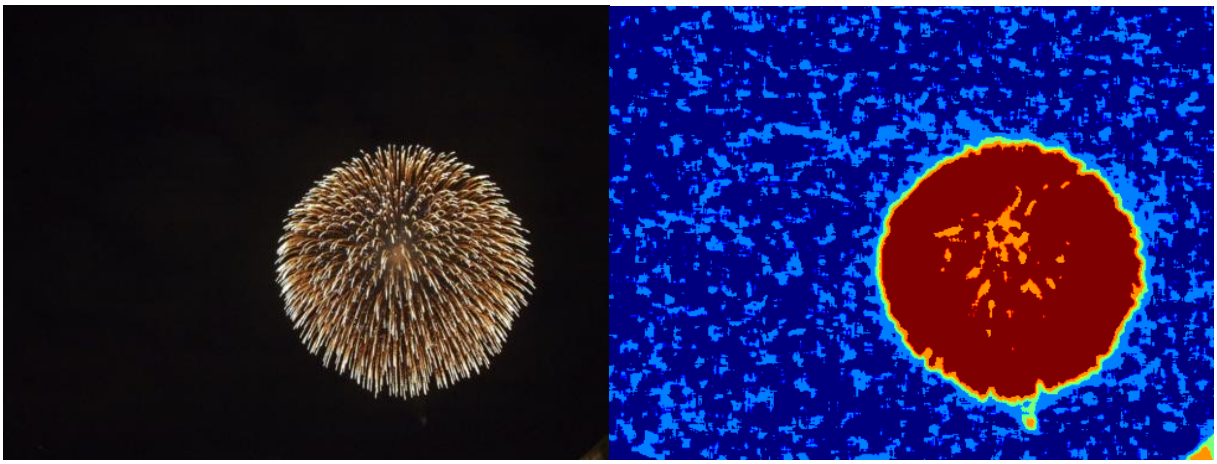


Abbildung 8 Beispiel für den Informationsgehalt eines Bildes

Da vor allem in den in Kapitel 3.2 als besonders problematisch festgestellten Bildtypen der Informationsgehalt sich stark zwischen den qualitativ relevanten Stellen (hier dem Feuerwerk) und den nicht relevanten Stellen (hier der schwarze Nachthimmel) unterscheidet, ist die Hypothese des Ansatzes, dass sich so richtige und falsche Ausschnitte differenzieren lassen.

Wird der Ausschnitt so gewählt, dass das Feuerwerk nicht enthalten ist, sinkt die Summe der enthaltenen Informationen im Vergleich zum Originalbild drastisch. Dieser Unterschied sollte sich detektieren lassen.

3.3.1 Methoden- und Konzeptentwicklung

Um den Informationsgehalt zu bestimmen, wird üblicherweise die Informations- oder auch Shannon Entropie [31] berechnet.

Die Shannon-Entropie eines beliebigen Ereignisses E berechnet sich durch den Logarithmus mit der Basis zwei der inversen Wahrscheinlichkeit dieses Ereignisses.

$$I_{(E)} = \log_2(\frac{1}{p_{(E)}})$$

Um die Entropie eines Bildes zu bestimmen, wird dieses zunächst in einen ein-dimensionalen Vektor umgewandelt und anschließend die pixel-weise Entropie berechnet, mit der jeweiligen Wahrscheinlichkeit multipliziert und schlussendlich aufsummiert. (siehe Abbildung 9)

```
def entropy(signal):
    prob = [n_x/len(signal) for x,n_x in collections.Counter(signal).items()]
    e_x = [-p_x * math.log(p_x, 2) for p_x in prob]
    return sum(e_x)
```

Abbildung 9: Python-Code zur Berechnung der Entropie eines 1D-Signals

Der so gewonnene Wert entspricht der durchschnittlichen Entropie des übergebenen Bildes.

Um die Informationsgehalte vergleichen zu können, müssen sowohl die Entropie des Originalbildes als auch die des Ausschnitts bestimmt werden.

Da sich die Entropie des Originalbildes nicht im Verlauf des Trainings ändert, reicht es aus diese einmal zu Beginn zu berechnen. Die Ausschnitte werden jedoch mehrfach im Training neu gewählt und ihr Informationsgehalt muss daher während des eigentlichen Trainings berechnet werden.

Der eigentliche Vergleich der Informationsgehälter kann über die Bildung einer einfachen Differenz durchgeführt werden. Da davon ausgegangen wird, dass sich das Motiv klar gegen den Rest des Bildes abhebt, sollte es deutlich mehr Information beinhalten und somit eine höhere Entropie haben.

Abbildung 10 zeigt ein Beispiel mit Originalbild und 3D-Entropieverteilung. Auf dem Entropiebild sind ein guter zufälliger Ausschnitt (grün) und ein schlecht gewählter Ausschnitt (rot) dargestellt.

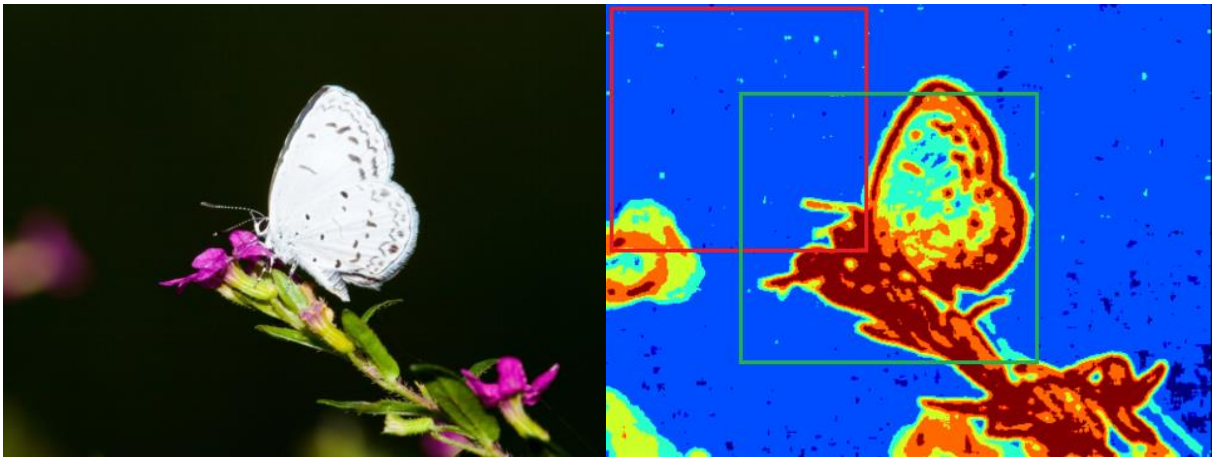


Abbildung 10 Entropiebeispiel mit Random Crops

Da das Motiv eine höhere Entropie aufweist, kann ein Grenzwert für die Differenz zum Originalbild festgelegt werden. Ausschnitte, die diesen Wert überschreiten werden nicht zum Trainieren des Netzwerks genutzt und verworfen.

Um diesen Grenzwert sinnvoll ermitteln zu können, und um die Anwendbarkeit des Ansatzes überhaupt festzustellen, wird zunächst die Verteilung der Entropie für verschiedene Ausschnitte geprüft.

Da bereits in Kapitel 3.2 erarbeitete wurde, in welchem Umfang und bei welcher Art Bilder das Problem üblicherweise vorkommt, wird hierfür ein extra Datensatz zusammengestellt. Dieser besteht aus ausgewählten Bildern, die zuvor als besonders anfällig eingestuft wurden.

Die Größe des Datensatzes besteht aus 30 individuellen Bildern, die sich in 15 Bilder mit großen hellen Flächen und 15 Bilder mit großen dunklen Bereichen aufteilen.

Beispiele dafür sind in Abbildung 11 gezeigt.

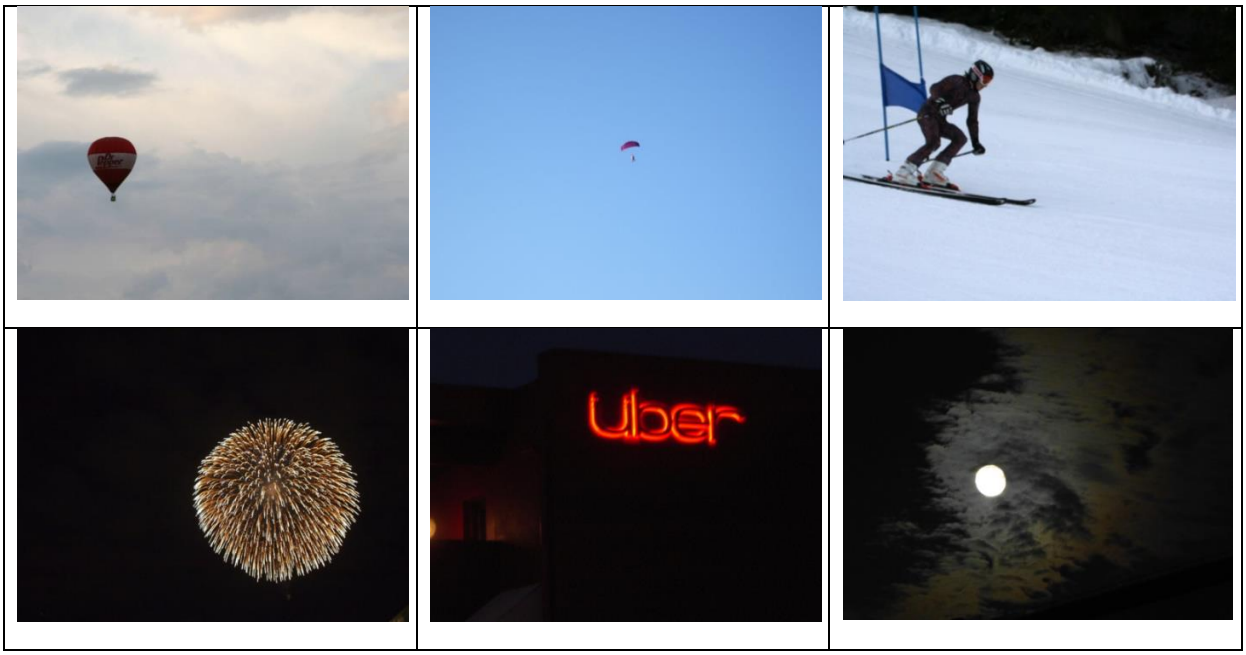


Abbildung 11 Auszüge der Testbilder

Zusätzlich werden von jedem Bild 5 zufällige Ausschnitte erstellt.

Diese Ausschnitte werden manuell überprüft und in zwei Gruppen eingeordnet. Ausschlaggebend ist, ob ein Großteil des qualitätsgebenden Motives des Bildes innerhalb der Auswahl liegt, oder nicht. Liegt ein großer Teil des Motives innerhalb des Bildausschnittes, wird es als positiver Treffer gewertet, andernfalls als negativer.

Die Verteilung der Entropiedifferenzwerte für positiven Treffer ist in Abbildung 12 dargestellt. Abbildung 13 zeigt die Verteilung für negative Treffer.

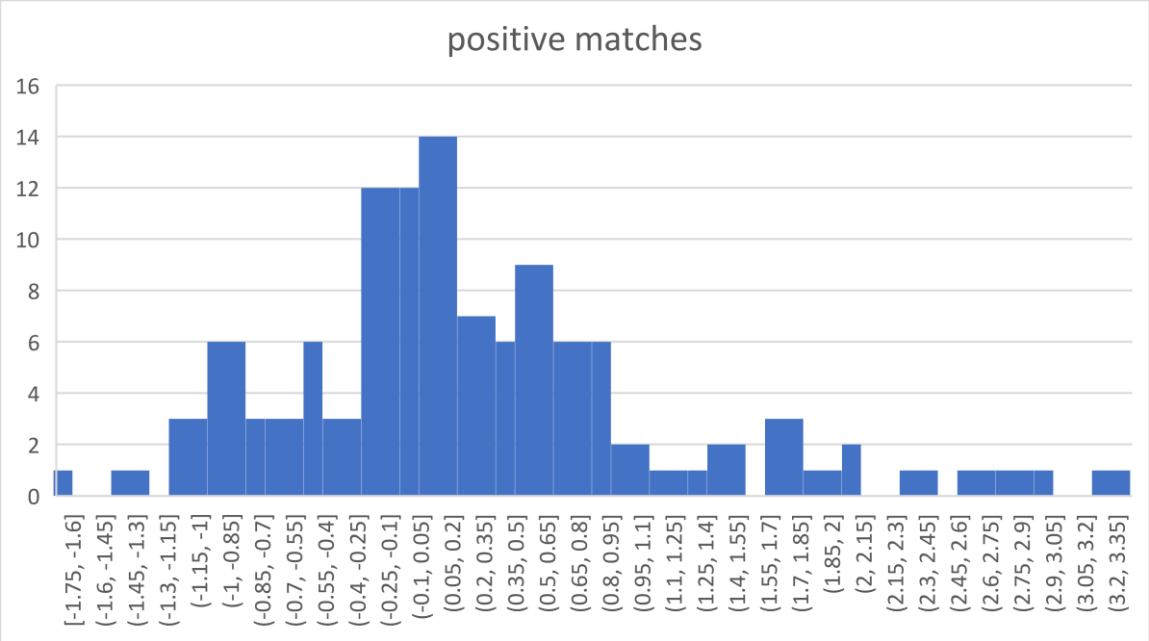


Abbildung 12 Verteilung der Entropiedifferenz für positive Treffer

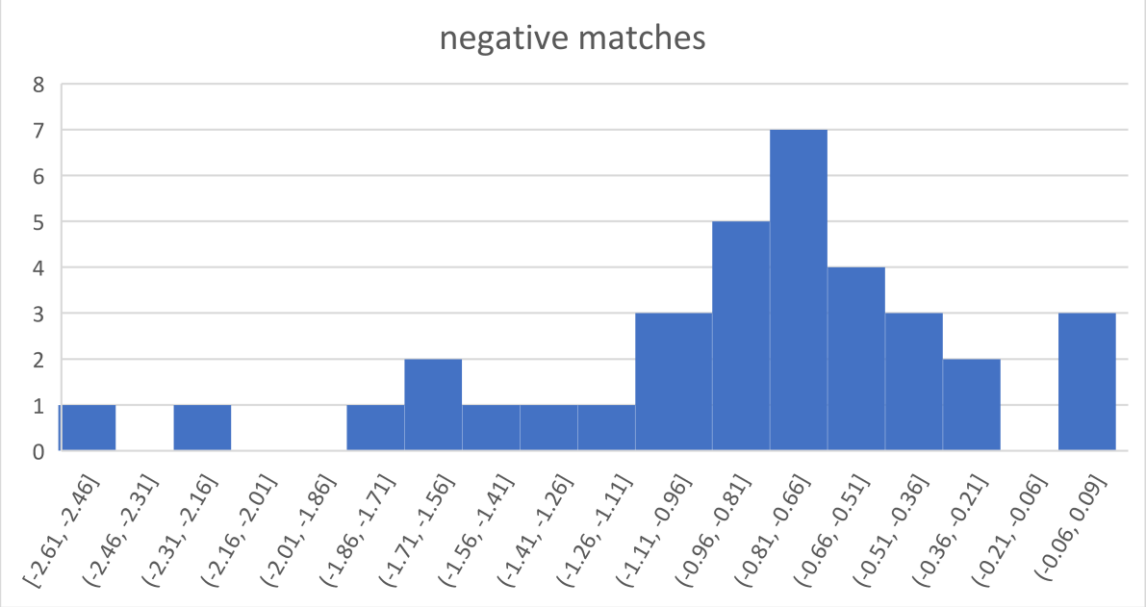


Abbildung 13 Verteilung der Entropiedifferenz für negative Treffer

Bei einem Vergleich der Verteilungen fällt schnell auf, dass zwischen positiven und negativen Entropiedifferenzwerten unterschieden werden muss. Dazu wird die Entropiedifferenz als $Entropie\ des\ Ausschnittes - Entropie\ des\ Gesamtbildes$ definiert.

Beide Verteilungen entsprechen grob einer Normalverteilung, unterscheiden sich aber in ihrem Erwartungswert. Dieser liegt bei repräsentativen Bildausschnitten etwa bei einer Entropiedifferenz von 0, bei nicht repräsentativen Ausschnitten aber bei etwa -0.7.

Weiterhin wird das Verhältnis von fehlerhaften Ausschnitten zu allen Ausschnitten in einem bestimmten Entropiedifferenzbereich ermittelt. Abbildung 14 zeigt dieses Verhältnis.

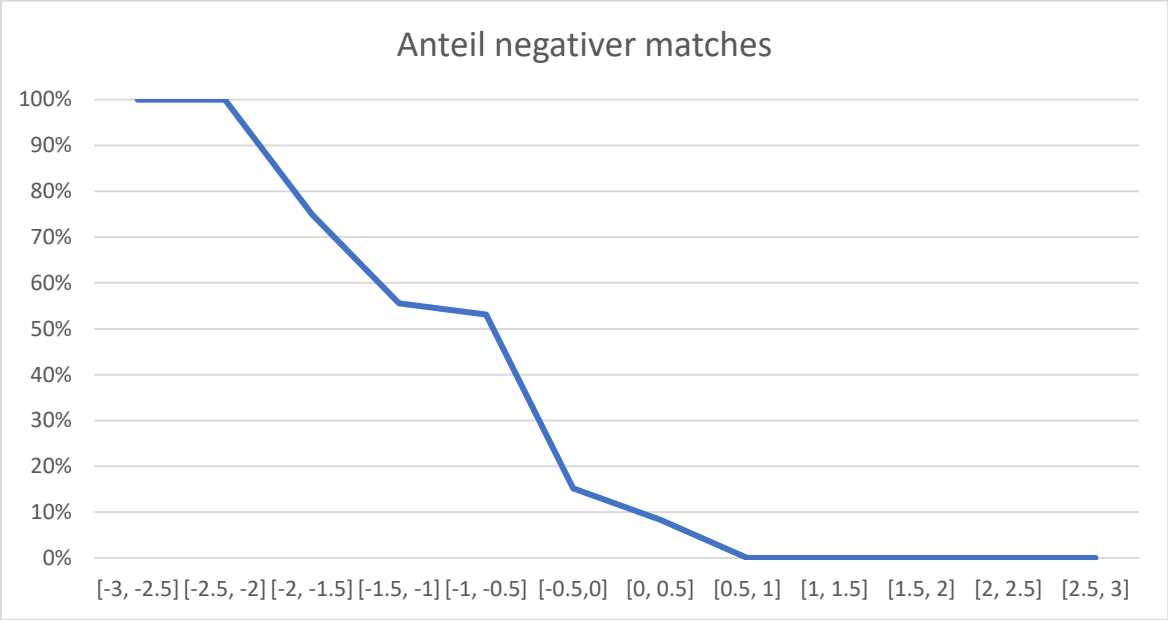


Abbildung 14 Anteil der negativen Matches nach Entropiedifferenz

Wie zu sehen ist, sind zwar die beiden Extrema (sehr hohe und sehr niedrige Entropiedifferenzen) klar zuzuordnen, aber der Mittelbereich zeigt einen annährend linearen Verlauf zwischen positiven und negativen Matches. Der Mittelpunkt liegt etwa bei einer Entropiedifferenz von ~ -1.

Daraus folgt bedeutet, dass beim Erhöhen des Grenzwertes, um mehr falsche Ausschnitte herauszufiltern, ebenso der Anteil fälschlicherweise gefilterter Ausschnitte steigt.

Um diesen Anteil so gering wie möglich zu halten, aber dennoch einen großen Teil der fehlerhaften Ausschnitte filtern zu können, wird der Grenzwert auf -0.5 festgelegt.

Eine weitere Notwendigkeit ist die Bestimmung, welcher Anteil der nicht-repräsentativen Ausschnitte so insgesamt erfasst wird.

Dazu wird das gleiche Verfahren, wie in Kapitel 3.2 genutzt. Die Probanden werden gebeten die Ausschnitte in qualitativ repräsentativ und nicht-repräsentativ einzuordnen. Zusätzlich werden die Entropie des Originalbildes und die des Bildausschnittes erfasst.

Abbildung 15 zeigt die Verteilung der Entropiedifferenz aller Bildausschnitte als Histogramm. Auch hier ist die annährend normale Verteilung um eine Differenz von 0 erkennbar.

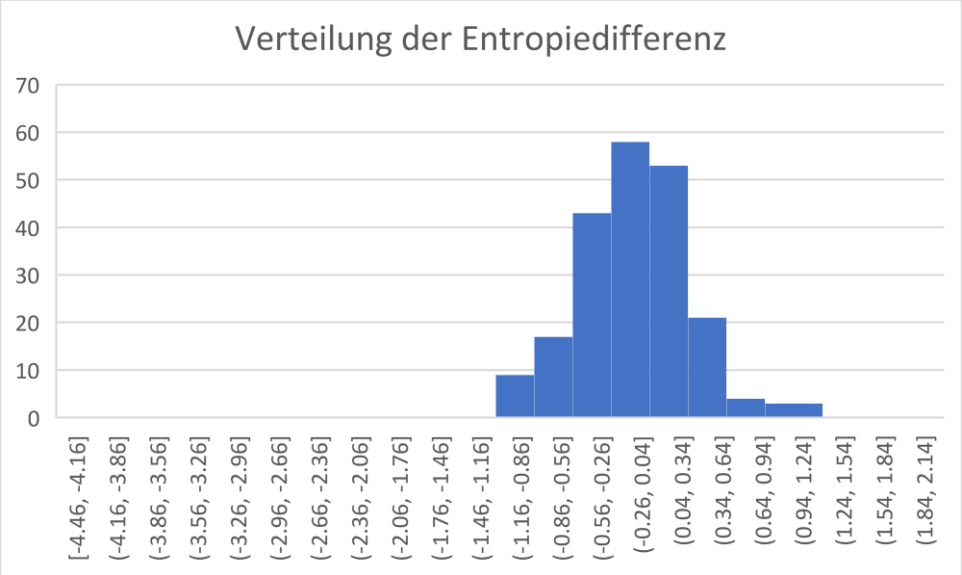


Abbildung 15 Verteilung der Entropiedifferenz

Insgesamt fallen 50% aller als nicht repräsentativ eingeschätzten Ausschnitte unter einen Differenzwert von -0.5.

3.3.2 Auswertung des Entropievergleichs

Die Auswertung des Ansatzes erfolgt unter drei Gesichtspunkten:

- Geschwindigkeit
- Genauigkeit
- Fehlerrate

Die Geschwindigkeit wird zwar als sekundäres Kriterium erachtet, ist aber trotzdem wichtig, um eine Aussage über den Reproduktionsaufwand und die im zeitlich begrenzten Rahmen der Arbeit möglichen Tests zu treffen.

Als Genauigkeit wird der Anteil der insgesamt filterbaren, nicht repräsentativen Ausschnitte bezeichnet.

Die Fehlerrate quantifiziert, wie viele Ausschnitte, deren Inhalt eigentlich qualitativ repräsentativ wäre, nicht zum Training des Netzwerks genutzt werden würden.

Geschwindigkeit

Zur Berechnung der Geschwindigkeit wird die mittlere Rechendauer aus ~ 10000 Berechnungen der Entropie von Bildausschnitten der Größe 224x224 Pixel ermittelt.

Die Größe von $224x224$ Pixeln entspricht der Eingangsgröße vieler gängiger Netzwerke zur Verarbeitung von Bildern, unter anderem ResNet, welches als Basis von TReS und vielen anderen IQA-Netzwerke genutzt wird.

Die Geschwindigkeit beträgt hier: $\sim 30.36 ms$

Um insgesamt Rechenleistung sparen zu können, werden die Entropiewerte der Gesamtbilder vor Beginn des Trainings berechnet. Die dafür notwendige Rechenleistung beträgt etwa $\sim 300 s$ und ist im Vergleich zum gesamten Trainingsaufwand des Netzwerks vernachlässigbar.

Der zeitliche Mehraufwand bei einer Trainingsgeneration entspricht mit der Testhardware etwa 204 Minuten. Für einen vollständigen Trainingsdurchlauf mit 9 Generationen und einer Batch-size von 40, werden etwa 63 Stunden benötigt.

Genauigkeit

Der Anteil der Ausschnitte, die als nicht-repräsentativ erachtet werden und somit aussortiert werden sollen liegt insgesamt bei etwa $6{,}3\%$. Der filterbare Anteil davon ist stark von anderen Faktoren abhängig. Beispielsweise, wie bereits in Kapitel 3.3.1 erläutert, steigt die Fehlerrate annähernd linear zur Genauigkeit.

Daher ist diese immer ein Kompromiss.

Mit dem gewählten Grenzwert für die maximale Entropiedifferenz eines Bildausschnittes von $-0{,}5$, werden in etwa 77% der nicht-repräsentativen Ausschnitte erfasst.

Dabei ist aber anzumerken, dass durch den relativ geringen Anteil der nicht-repräsentativen Ausschnitte an den gesamten Ausschnitten Stichprobengröße eher klein und die Varianz des Ergebnisses damit hoch ist.

Fehlerrate

Die Fehlerrate ist zwar kein primäres Kriterium, um die Leistung des Lösungsansatzes zu bewerten, ist aber trotzdem sehr wichtig, um die Funktionalität einordnen zu können und um mögliche Konsequenzen für das Gesamtsystem abzuschätzen.

Das primäre Ziel ist es zwar den Anteil fehlerhafter Trainingsdaten zu verringern, wenn jedoch durch eine zu hohe Fehlerrate die Größe und Diversität des Datensatzes zu stark eingeschränkt werden, können negative Konsequenzen für die Robustheit und Fähigkeit zu Generalisieren des Gesamtsystems entstehen.

Wie bereits erwähnt und in Abbildung 14 dargestellt, ist die Rate der fälschlich aussortierten Bildausschnitte stark mit der Genauigkeit verbunden. Bei dem gewählten Entropiedifferenzgrenzwert von -0,5 beträgt diese Rate etwa 47%.

Dies bedeutet, dass annähernd die Hälfte aller Ausschnitte zwar eine qualitative Repräsentation des Originalbildes ist, aber dem Trainingsdatensatz nicht zur Verfügung steht.

3.4 Ansatz 2: Feature Matching

Eine weitere häufig genutzte Methode zum Vergleich zweier Bilder ist das sogenannte *Feature Matching*. Dabei werden zunächst mehrere signifikante Merkmale sowohl aus dem Originalbild als auch dem Bildausschnitt extrahiert. Anschließend wird geprüft, ob sich diese Merkmale in beiden Bildern finden lassen.

Die Annahme ist, dass die Qualität eines Bildes hauptsächlich durch das Motiv bestimmt wird und, dass das Motiv im Regelfall das erkennbarste Objekt im Bild ist.

Sollte ein größerer Anteil der extrahierten Merkmale in beiden Bildern vorkommen, kann davon ausgegangen werden, dass beide Bilder dasselbe Motiv oder dieselbe Szene darstellen. Solche Merkmale sind üblicherweise Kanten, die sich farblich abheben.

Dies ist exemplarisch in Abbildung 16 dargestellt. In diesem und den folgenden Beispielbildern ist das Originalbild links und der Ausschnitt oben rechts dargestellt. Die eingefärbten Kreise zeigen erkannte Merkmale und die Linien zeigen die Zuordnung.

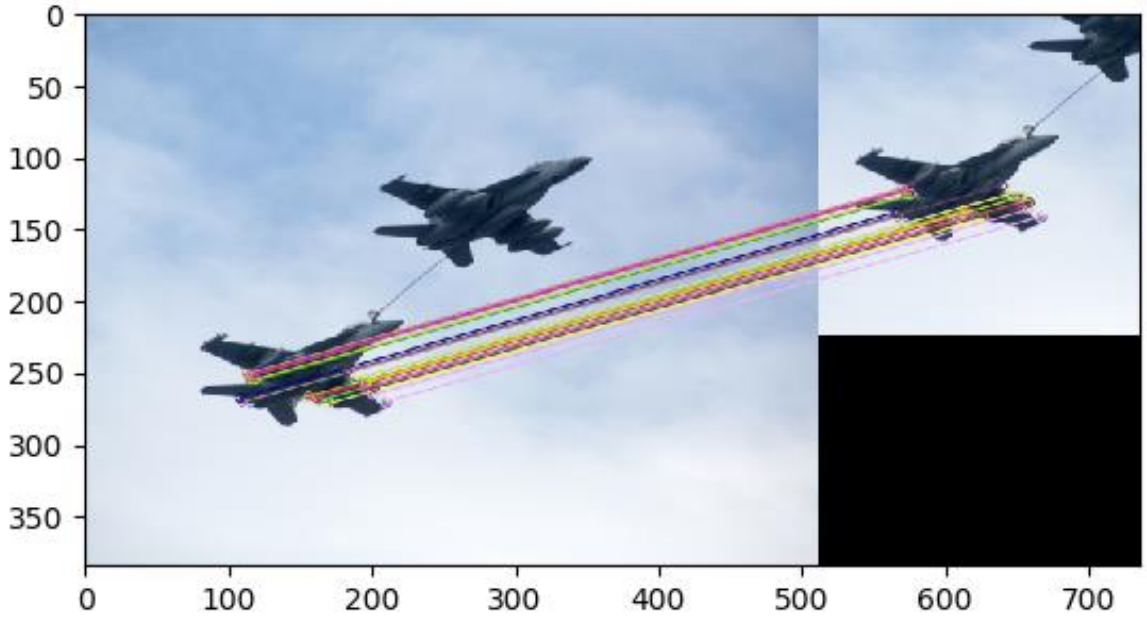


Abbildung 16 Feature Matching eines Flugzeugs

3.4.1 Methoden- und Konzeptentwicklung

Der Ansatz hier ist, dass davon ausgegangen werden kann, dass im Regelfall das Motiv eines Bildes die meisten Merkmale aufweist und diese klar zuzuordnen sind. Die Qualität der Zuordnungen, beziehungsweise die Ähnlichkeit, werden über die “Distanz” angegeben. Je niedriger die Distanz, desto besser ist die Übereinstimmung.

Auch dieser Ansatz beruht, ähnlich wie der Entropievergleich, darauf dass, vor allem in den problematischen Fällen, die für die Qualität der Bilder ausschlaggebenden Elemente einen hohen Erkennungswert haben.

Die Bildausschnitte fallen in zwei verschiedenen Kategorien.

Zum einen müssen genug Merkmale vorhanden sein, um eine plausible Aussage über das Bild treffen zu können. Ist beispielsweise nur der eintönige Hintergrund im Bildausschnitt sichtbar, können teilweise keine oder nur uneindeutige Merkmale festgestellt werden.

Eindeutig zuordenbare Merkmale haben eine Distanz von 0.

Ein Beispiel dafür ist Abbildung 17, in der lediglich 10 Merkmale zugeordnet werden konnten. Keines dieser Merkmale weißt du eine Distanz von 0 auf.

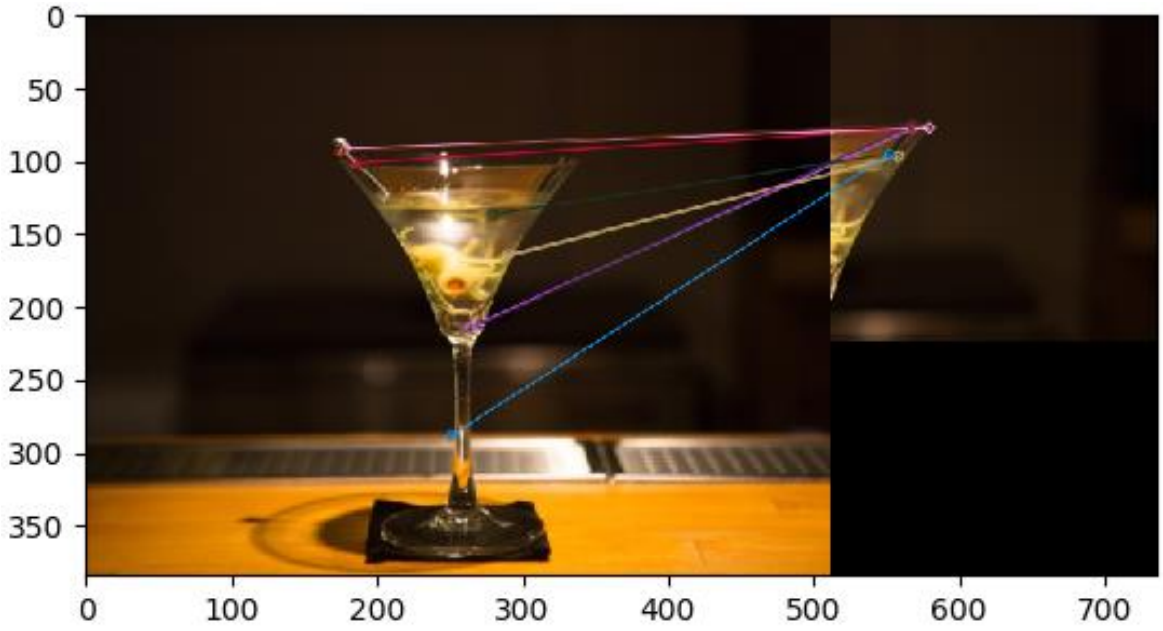


Abbildung 17 Fehlgeschlagenes Feature Matching eines Cocktailglases

Der zweite Fall ist, dass im Bild eindeutig zuordenbaren Merkmale erkannt werden. Abbildung 18 zeigt den Vergleich zu Abbildung 17. Hier wurdenAbbildung 18 182 Merkmale zugeordnet, von denen 27 eine Distanz von 0 haben.

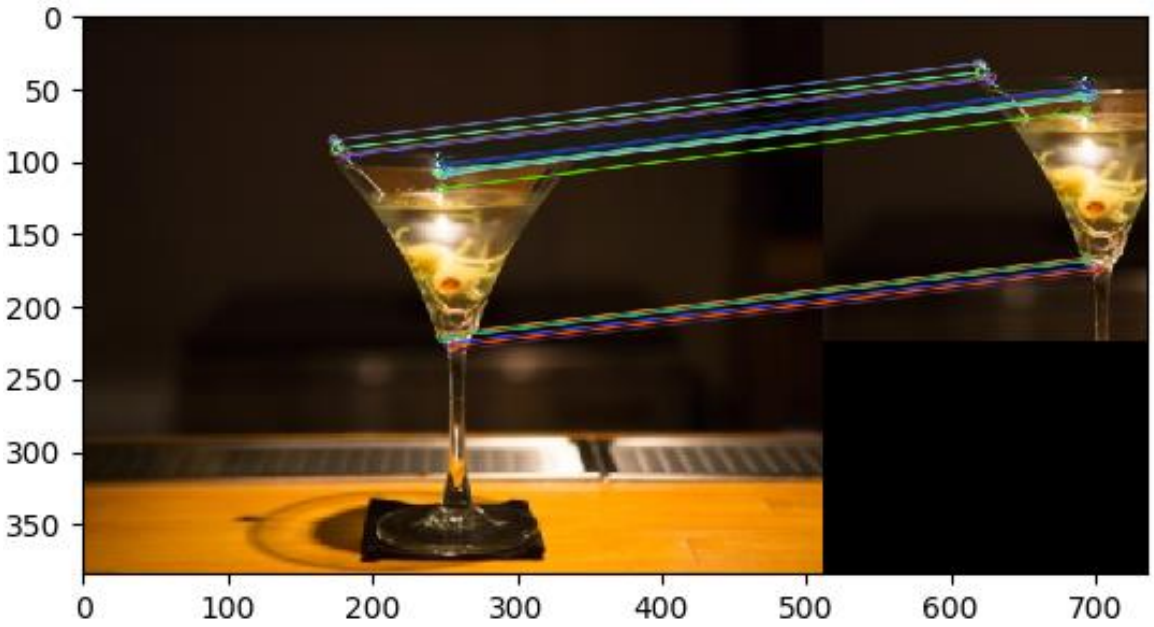


Abbildung 18 Besseres Feature Macthing des Cocktailglases

Da die meisten Bilder irgendeine Art von zuordenbaren Merkmalen beinhalten und somit unter diese Kategorie fallen, ist es notwendig hier ein Kriterium zu schaffen, dass in der Lage ist aus dieser Information eine zuverlässige Entscheidung über die Validität des Ausschnittes zu treffen.

Das Feature Matching wird zunächst implementiert, um mögliche Ansätze und Lösungen zu prüfen.

Um die Implementierung zu vereinfachen, wird zur Erprobung die in OpenCV bereitgestellte Klasse $Feature2D()$ [32] genutzt. Die Klasse beinhaltet mehrere Möglichkeiten zum Erstellen der Merkmale. Basierend auf den in [33] angestellten Vergleichen der verschiedenen Verfahren zur Gewinnung und zum Vergleich der Merkmale, werden jedoch nur zwei vielversprechende Kombinationen aus Feature-Extraktor/Feature-Matcher getestet. Diese sind:

- SURF/BF
    -
- ORB/BF
    -

Die beiden Kategorien werden separat behandelt.

Zuerst werden Fälle, die unter die erste Kategorie fallen, behandelt.

Diese Fälle, in denen keine Merkmale entweder nur im Originalbild, nur im Bildausschnitt oder in beiden Bildern bestimmt werden konnten, werden nachfolgend als “Fehler“ bezeichnet.

Um die Fehler dieses Typs zu quantifizieren, wird zunächst überprüft welcher Anteil aller Ausschnitte in diese Kategorie fällt. Dazu werden ~ 100000 Bildausschnitte mit beiden Feature-Matching-Kombinationen geprüft. So ergeben sich Raten von:

- SURF/BF: $2{,}26\%$
- ORB/BF: $3{,}58\%$

Sie lassen sich ein zwei Gruppen einordnen:

- Es sind keine Merkmale im Originalbild oder im Original- und im Teilbild vorhanden
    - Annahme: Das Bild beinhaltet nicht genug markante Stellen, um ein Feature Matching durchzuführen
- Es werden Merkmale im Originalbild erkannt, aber nicht im Teilbild
    - Annahme: Die im Bild enthaltenen Merkmale sind bei der Wahl des Ausschnitts vermutlich nicht erfasst worden.

Der erste Fall ist in Abbildung 19 und Abbildung 20 dargestellt. Der Anteil dieser Fehlerart an den gesamten Fehlern entspricht bei SURF/BF $5{,}7\%$. und bei ORB/BF $1{,}6\%$.

Alle Fehlerfälle dieser Art treten in weitgehend einfarbigen Bildern auf, in denen entweder kein klares Motiv vorhanden ist (siehe Abbildung 19), oder das Motiv keine einzigartigen Stellen (Ecken) besitzt, die eine Zuordnung möglich machen würden (siehe Abbildung 20).

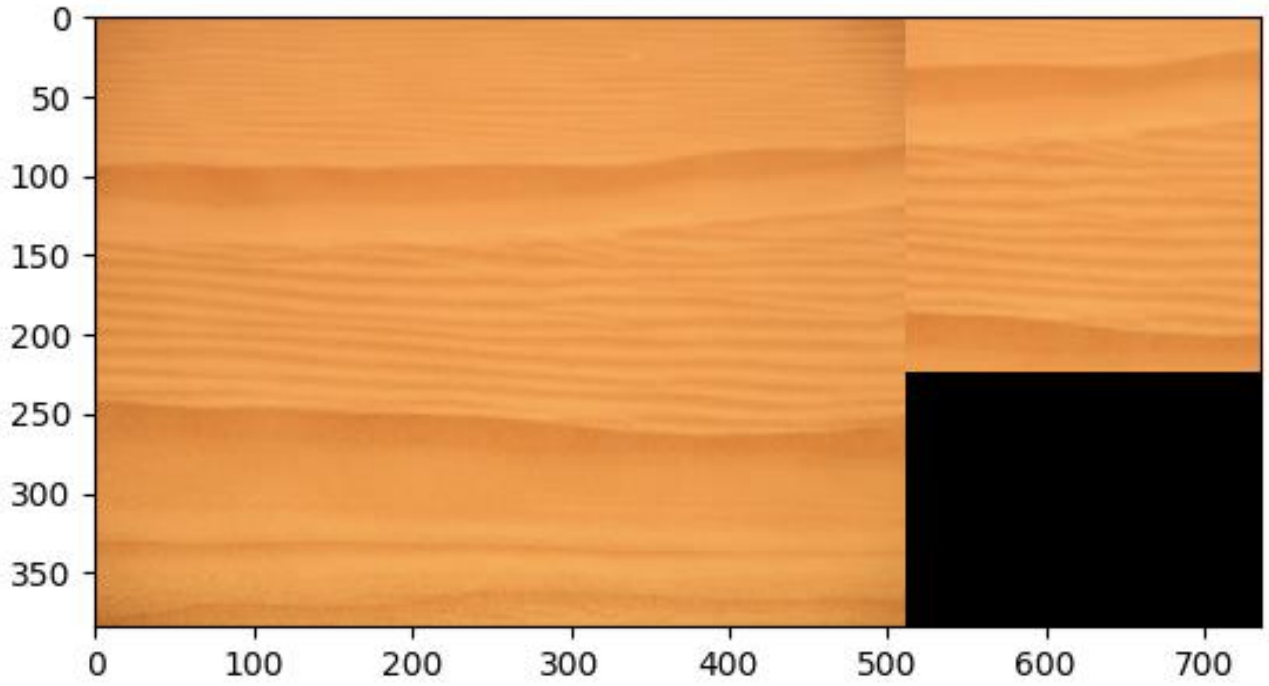


Abbildung 19 Wüste ohne erkennbare Merkmale

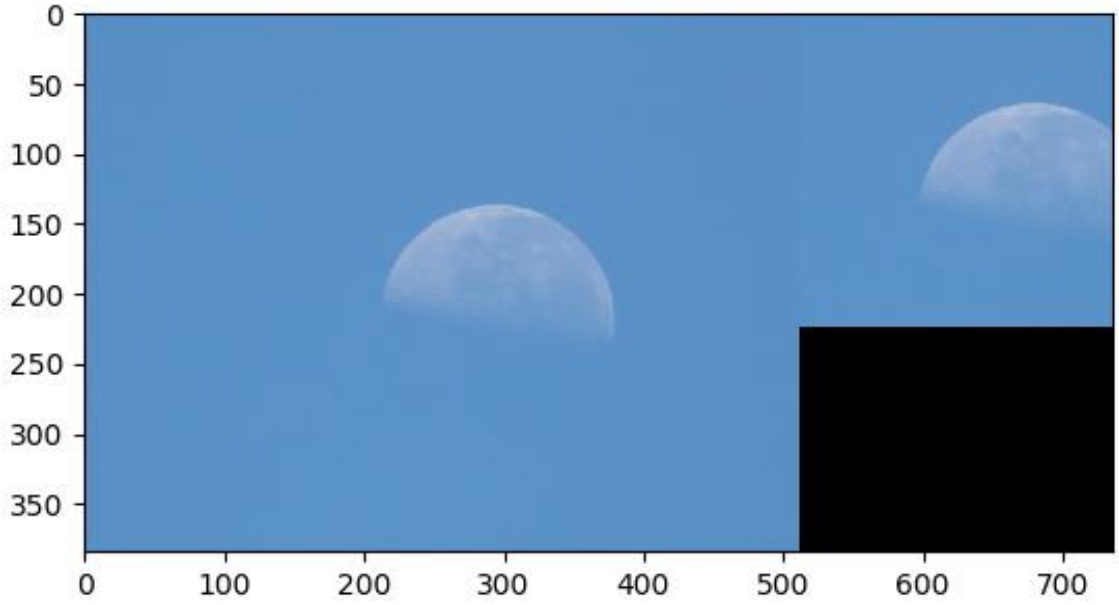


Abbildung 20 Mond zur Tageszeit ohne erkennbare Merkmale

Da der erste Fall nur einen kleinen Anteil der Fehler darstellt und sich oftmals äußert, wenn kein klares Motiv vorhanden ist, sondern das ganze Bild, oder der größte Teil des Bildes, qualitativ relevant ist, können Ausschnitte, die Fall 1 entsprechen als gültiges Bild gewertet werden.

Der zweite Fall ist in Abbildung 21 zu sehen. Hieraus setzt sich der deutlich größere Anteil der Fehler zusammen, mit 94,3% bei SURF/BF und 98,4% bei ORB/BF.

Abbildung 21 Bildausschnitt beinhaltet keine Merkmale

In diesem Fall sind zwar markante Merkmale im Originalbild vorhanden und werden erkannt, der zufällige Ausschnitt ist jedoch so gewählt, dass dort keine oder nur wenige einzigartigen Merkmale erkannt werden können. Somit kann auch kein Objekt eindeutig zugeordnet werden.

Fall 2 tritt meistens auf, wenn zwar ein klares Bildmotiv vorhanden ist, der Hintergrund aber so homogen ist, dass dort keine Merkmale festgemacht werden können. Liegt der zufällige Bildausschnitt nun abseits des Motives, können keine Merkmale ausgemacht werden. Ein weiteres Beispiel hierfür ist in Abbildung 22 gegeben.

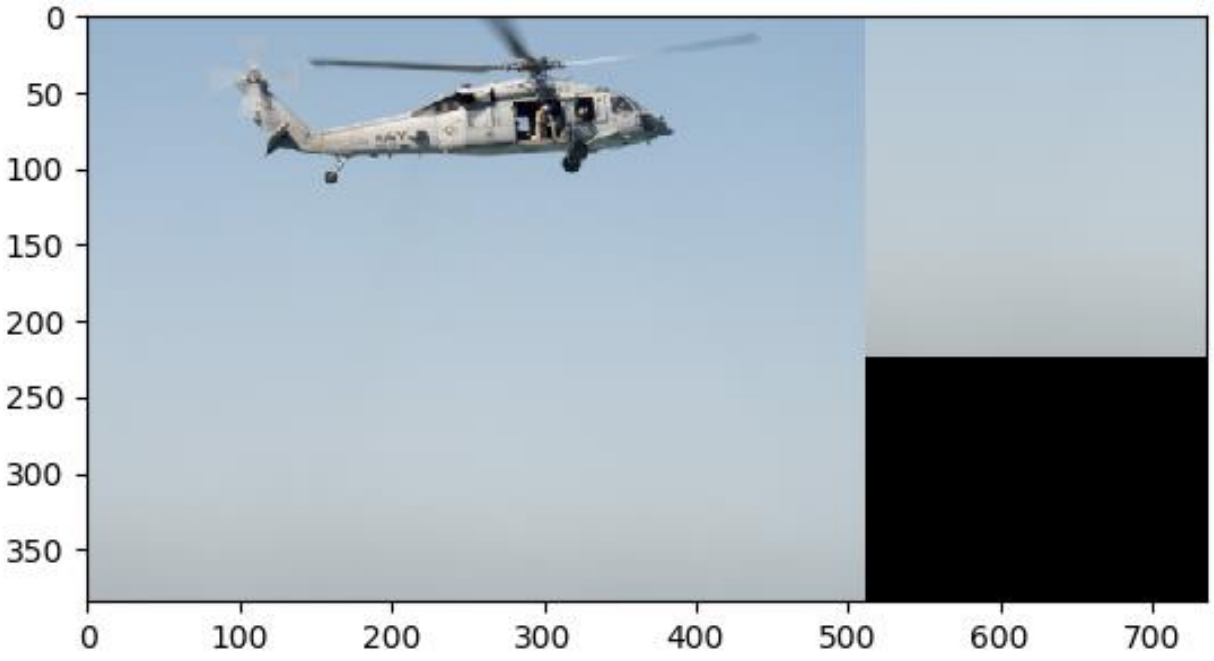


Abbildung 22 Bildausschnitt Helikopter

Bei einer manuellen Sichtung der Fehler des zweiten Falles lassen sich diese nochmals in 5 verschiedene Gruppen (siehe Abbildung 23) einordnen:

- Korrekt (Abbildung 23a)
  - Fehler, bei denen das Bildmotiv, wie zu erwarten, durch die Wahl des Ausschnitts verloren geht.
- Form (Abbildung 23b)
  - Fehler die durch eine nicht zuordenbare Form des Motives entstehen. Ein häufiges Problem sind zum Beispiel Mondaufnahmen, die $\sim 36\%$ dieser Gruppe ausmachen.
- Entsättigung (Abbildung 23c)
  - Diese Gruppe beinhaltet Bilder, bei denen aufgrund einer Farbentsättigung nicht genug farbliche Unterschiede für eine Detektion von Merkmalen vorhanden sind. Eine solche Entsättigung kann durch schlechte Belichtung, nicht genug farbliche Variation, aber auch beispielsweise durch schlechte Unterwasseraufnahmen entstehen.
- Abstrakt (Abbildung 23d)
  - Unter Abstrakt werden Bilder zusammengefasst, die kein klares Motiv haben, wie die in Abbildung 19 gezeigte Wüstenlandschaft oder deren Motiv keine klare Form hat, wie beispielsweise bewegte Langzeitbelichtungen.
- Kein Grund
  - Hier erfasste Ausschnitte fallen in keine der anderen Gruppen und zeigen keinen ersichtlichen Grund, warum sie nicht erkannt werden sollten.

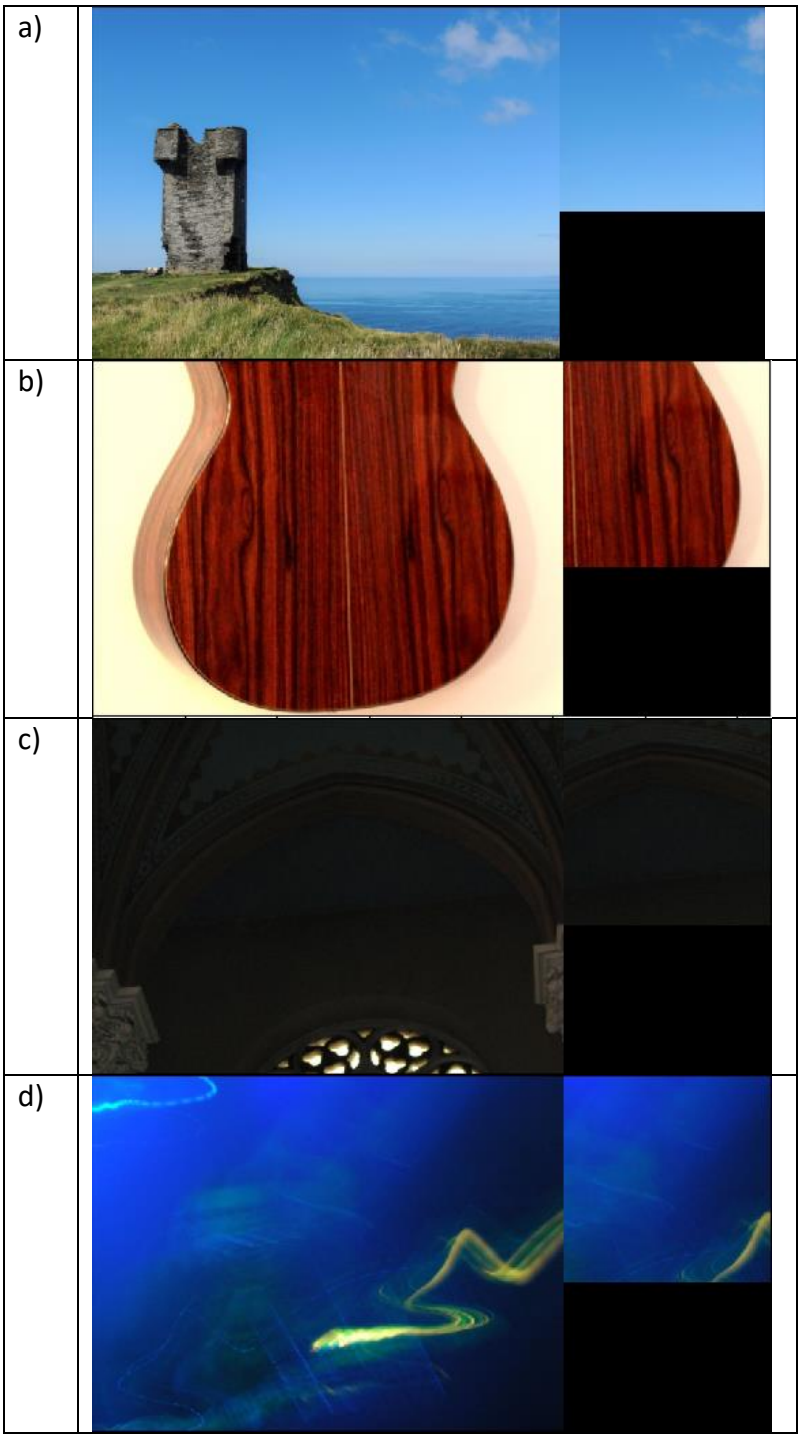


Abbildung 23 Feature Matching Fehlerkategorien

Abbildung 24 zeigt die Aufteilung der Gruppen anhand einer Sichtung von 217, beziehungsweise 367 solcher Fehlerfälle. Die Prüfung wurde sowohl für das System aus SURF und BF als auch ORB und BF durchgeführt.

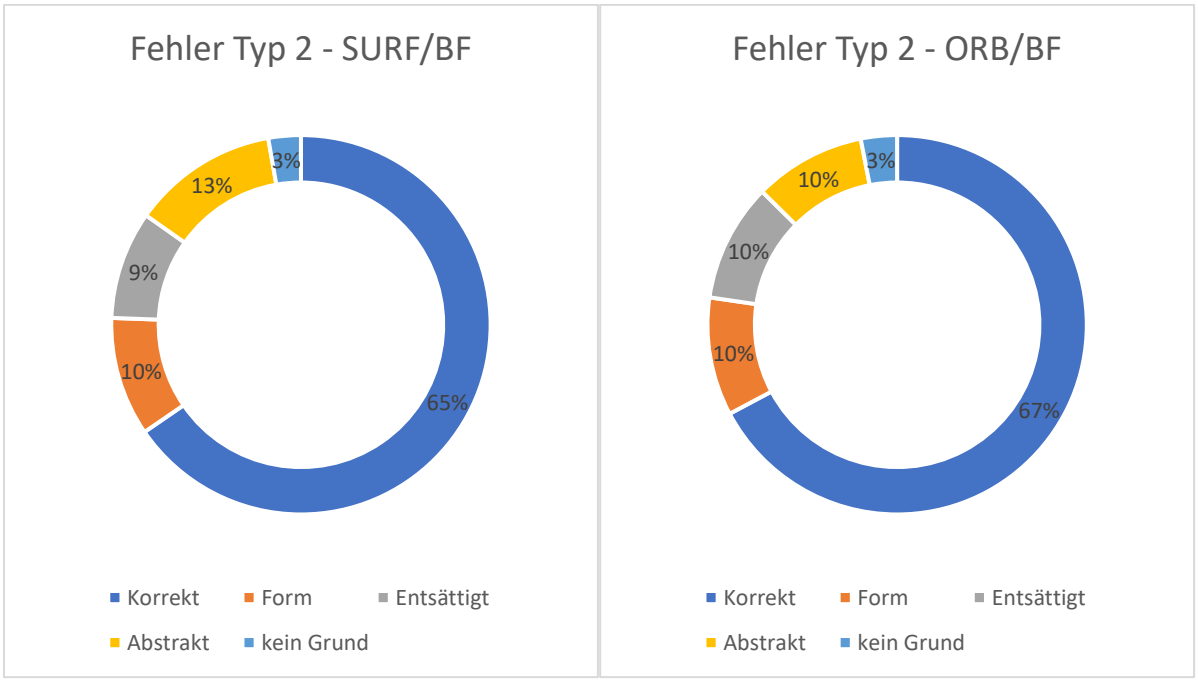


Abbildung 24 Zusammensetzung der Fehler des Typ 2

Wie zu erkennen ist, machen Fälle, die dem zu erwartenden Verhalten folgen, in denen also das Bildmotiv nicht mehr im Ausschnitt zu finden ist, mit etwa zwei Drittel den größten Anteil aus. Das restliche Drittel verteilt sich auf die unerwarteten Fälle. Dabei bestehen keine nennenswerten Unterschiede zwischen den beiden Ansätzen.

Um Rechenzeit zu sparen und da der Anteil dieser Fälle insgesamt relativ gering ist, wird auf eine separate Detektion und Behandlung der einzelnen Fehlergruppen verzichtet und es wird davon ausgegangen, dass Fehler des zweiten Typs immer einen ungültigen Ausschnitt beinhalten.

Auch gibt es hierbei keine großen Unterschiede in der Genauigkeit zwischen den beiden Feature Matching-Verfahren, weshalb fortan Daten nur durch ein Verfahren gewonnen werden.

Die meisten Bilder enthalten jedoch in irgendeiner Form Detektionen, die zugeordnet werden können, und fallen somit in die zweite Kategorie. In dieser Kategorie sind sowohl im Originalbild als auch im Bildausschnitt Merkmale erkennbar, die auch einander zugeordnet werden können.

Der Ansatz hier ist eine Betrachtung der gefundenen Übereinstimmungen, indem der Anzahl der zugeordneten Merkmale mit einer Distanz von 0 mit den gesamten zugeordneten Merkmalen verglichen wird.

Die Hypothese ist, dass bei Fällen, in denen das Hauptmotiv nicht im Ausschnitt liegt, das Verhältnis von

$$\frac{Merkmale\ im\ Bildausschnitt}{Gesamte\ Merkmale}$$

geringer ist als bei regulären Bildausschnitten.

Um die Hypothese zu überprüfen, wird dieses Verhältnis für eine zufällige Auswahl an Ausschnitten bestimmt und die Ausschnitte werden in repräsentativ und nicht-repräsentativ eingeordnet. Die beiden Arten an Ausschnitten werden zunächst getrennt betrachtet.

Zuerst werden repräsentative Ausschnitte betrachtet.

Dazu wird das gleiche Verfahren, wie in Kapitel 3.2 genutzt. Die Probanden werden gebeten die Ausschnitte in qualitativ repräsentativ und nicht-repräsentativ einzuordnen. Zusätzlich werden die Merkmale und Zuordnungen erfasst.

Abbildung 25 zeigt die Verteilung des genannten Verhältnisses für diese Ausschnitte als Histogramm.

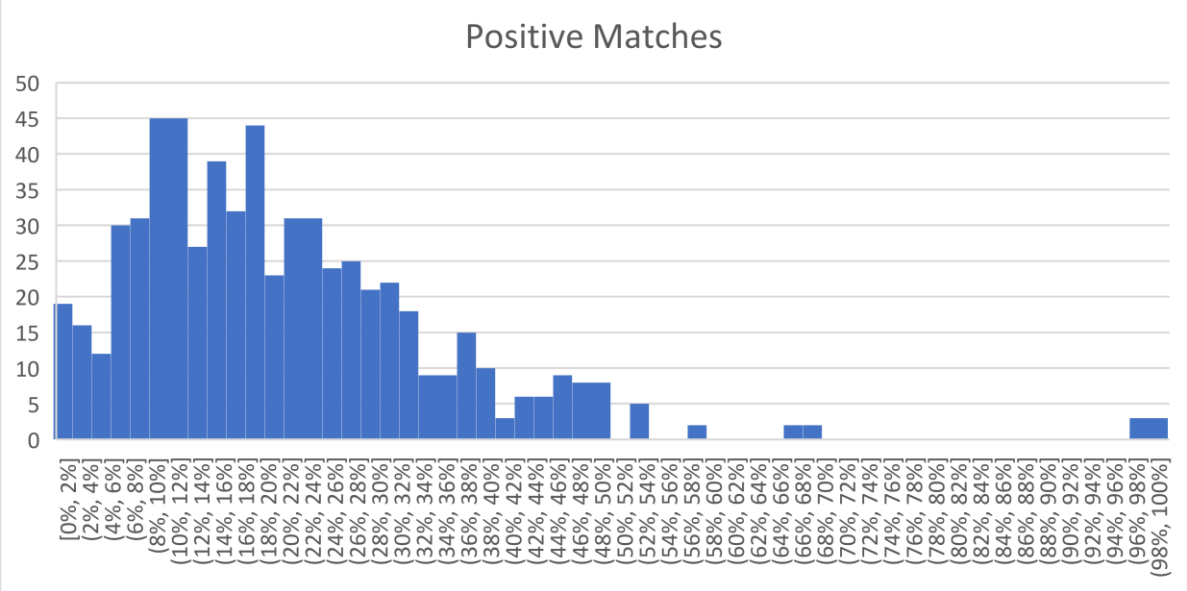


Abbildung 25 Histogramm der positiven Übereinstimmungen

Die X-Achse zeigt das Verhältnis $\frac{null-distance\ matches}{all\ matches}$ in Prozent für alle korrekten Bildausschnitte.

Nahezu alle Bildausschnitte, deren Inhalt das Bild qualitativ widerspiegelt, weisen ein Verhältnis von unter 50% auf. Die größte Mehrheit liegt dabei im Bereich von etwa 5% bis 35%.

Als nächstes werden die Fälle betrachtet, in denen der Inhalt des Ausschnittes nicht qualitativ repräsentativ für das Gesamtbild ist.

Die Verteilung dafür ist wieder als Histogramm in Abbildung 26 dargestellt.

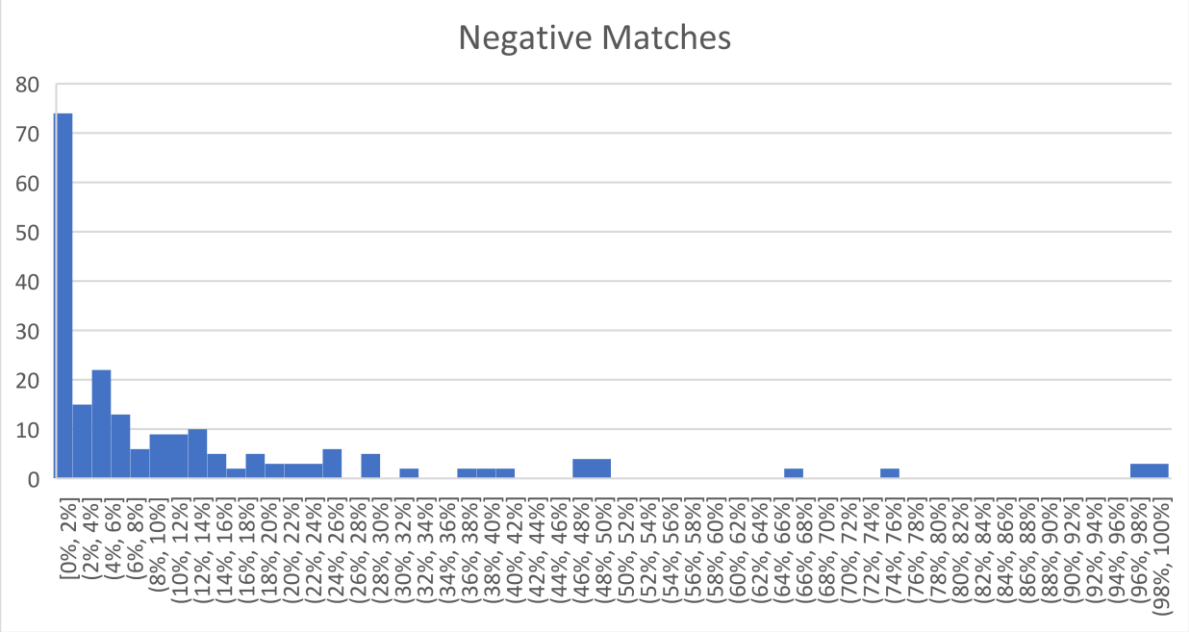


Abbildung 26 Histogramm der negativen Übereinstimmungen

Die Verteilung in diesem Fall ist deutlich einseitiger. Mehr als ein Drittel aller Ausschnitte, die hierunter fallen, weist ein Verhältnis von unter 2% auf. Etwa zwei Drittel aller Ausschnitte liegen unter einem Verhältnis von 10%.

Bei der Betrachtung beider Histogramme ist schnell erkennbar, dass lediglich die Spitze bei einem Verhältnis von unter 2% in Abbildung 26 als Ansatzpunkt sinnvoll genutzt werden kann. Diese wird daher näher betrachtet und mit dem korrespondierenden Verhältnisbereich aus Abbildung 25 verglichen.

Dabei fällt auf, dass ein großer Anteil der Ausschnitte, die darunterfallen, keine einwandfreien Zuordnungen besitzen und somit ein Verhältnis von 0% vorweisen.

Abbildung 27 zeigt die Verteilung aller Abschnitte mit einem Verhältnis von 0%.

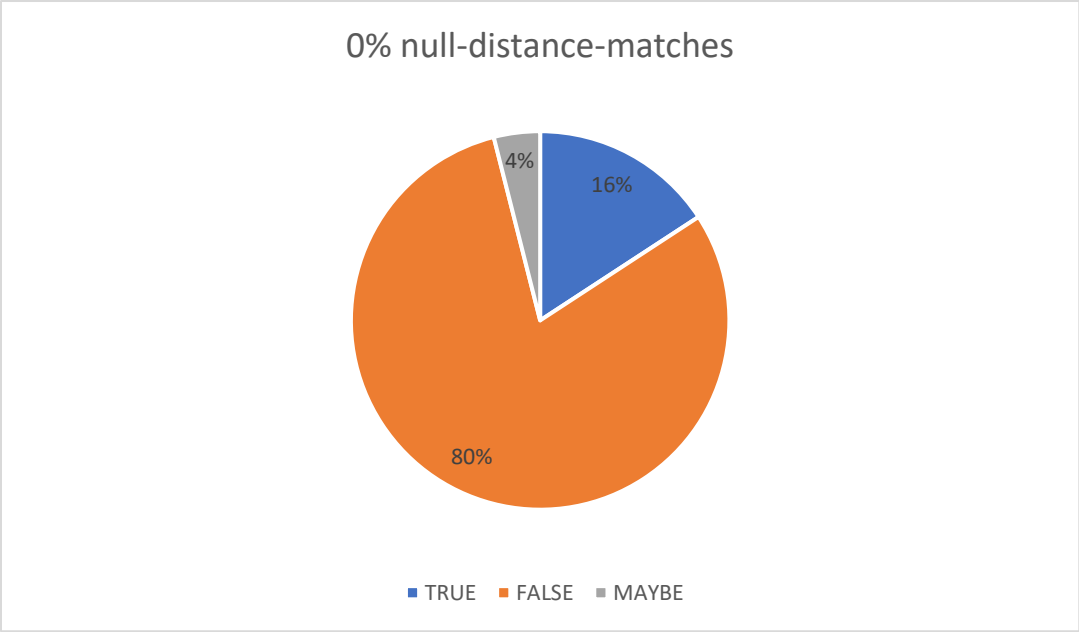


Abbildung 27 Verteilung für einen Feature-Matching Grenzwert von 0%

Der Anteil mit dem Label „TRUE“ bezieht sich auf Ausschnitte, die die Qualität des Gesamtbildes widerspiegeln. „FALSE“ beinhaltet die Ausschnitte, bei denen dies nicht der Fall ist. Das Label „MAYBE“ steht für Ausschnitte, die von den Probanden nicht einwandfrei zugeordnet werden konnten.

Wie zu erkennen ist, ist der überwiegende Anteil der Random Crops, der keine zuordenbare Merkmale mit einer Distanz von 0 aufweist, nicht repräsentativ für die Qualität des Bildes.

Als Vergleich ist in Abbildung 28 dieselbe Aufteilung für ein Verhältnis von kleiner als 2% dargestellt.

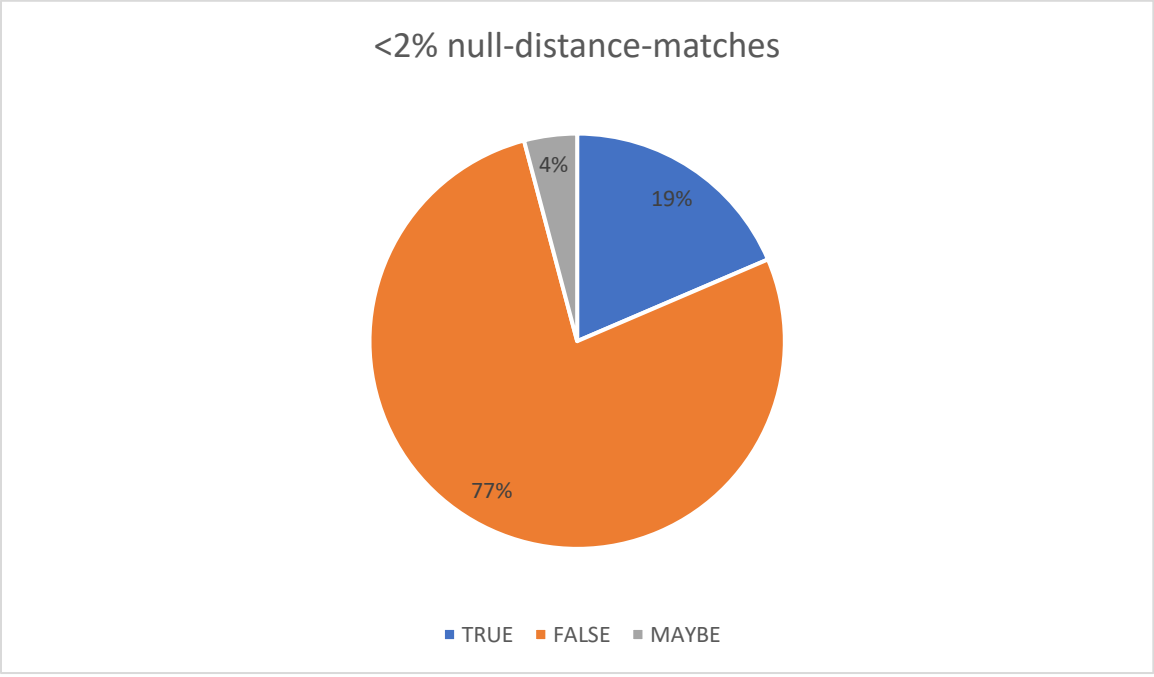


Abbildung 28 Verteilung für einen Feature-Matching Grenzwert von <2%

Die beiden Verteilungen unterscheiden sich nur unwesentlich in ihrer Zusammensetzung. Bei einer Erhöhung des Verhältnisses sinkt der Anteil nicht repräsentativer Ausschnitte von 80% auf 77% und der Anteil repräsentativer Crops erhöht sich im selben Maße.

Bei einem Verhältnis-Grenzwert von 0% sind insgesamt $29{,}8\%$ aller nicht repräsentativer Ausschnitte enthalten. Die Erhöhung auf 2% lässt diesen Wert auf $36{,}6\%$ steigen.

Tabelle 3 stellt die beiden Möglichkeiten direkt gegenüber.

Tabelle 3 Gegenüberstellung der Verhältnis-Grenzwerte

Verhältnis	% nicht repr.-Ausschnitte	% Ausschnitte insgesamt
0%	80%	29,8%
<2%	77%	36,6%

Da ein Grenzwert von unter 2% trotz seiner leicht erhöhten Rate an, fälschlicherweise erfassten, repräsentativen Ausschnitten, aber einen im Vergleich deutlich höheren Anteil, der nicht repräsentativen Ausschnitte aufweist, wird der Grenzwert auf $< 2\%$ festgelegt.

3.4.2 Auswertung des Feature Matchings

Die Auswertung des Feature Matching Ansatzes erfolgt unter denselben Gesichtspunkten wie in Kapitel 3.3.2. Diese sind die Geschwindigkeit, die Genauigkeit und die Fehlerrate. Die Auslegungen der einzelnen Punkte sind ebenfalls identisch.

So lässt sich die Vergleichbarkeit der beiden Lösungsansätze gewährleisten.

Geschwindigkeit

Die Berechnung der Geschwindigkeit erfolgt ähnlich zu Kapitel Kapitel 3.3.2. Es werden die Merkmale in ~ 100000 zufälligen Bildausschnitten mit einer Größe von 224x224 Pixeln detektiert und dem Originalbild zugeordnet und die durchschnittliche Rechenzeit ermittelt.

Die Dauer dieser Operation unterscheidet sich dabei stark zwischen den beiden erwogenen Feature-Matching-Ansätzen. Sie ist ist Abbildung 29 gezeigt.

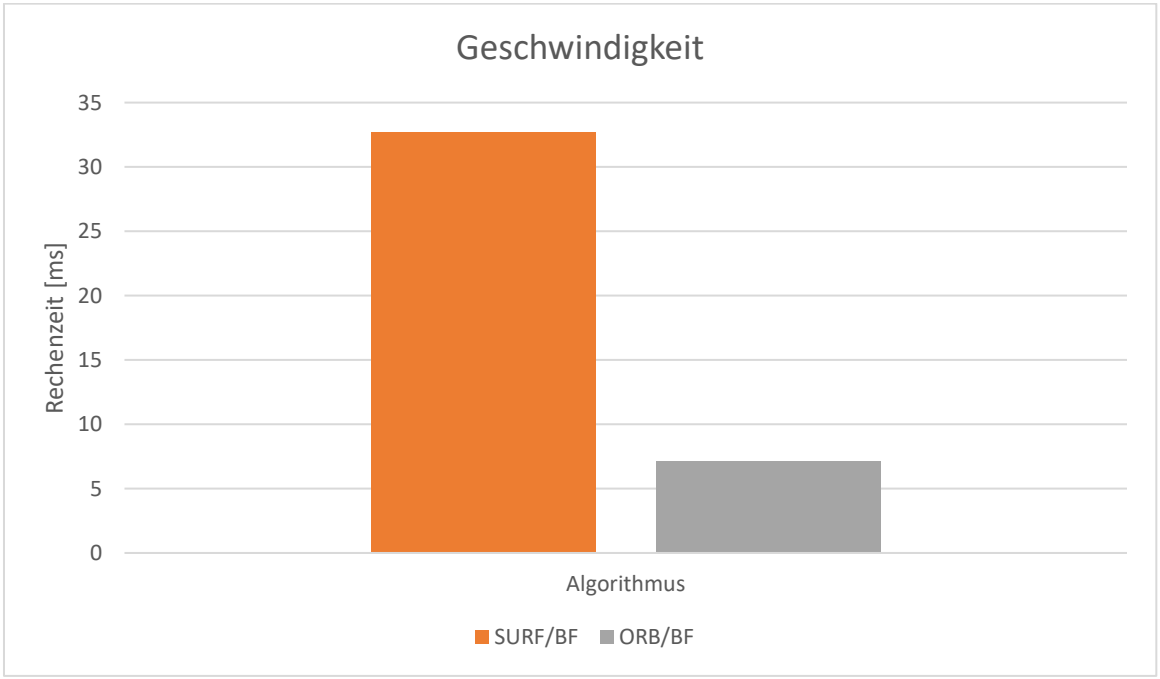


Abbildung 29 Geschwindigkeitsvergleich beim Feature Matching

Die Kombination aus SURF zum Erkennen der Merkmale und BF zum Zuordnen benötigt im Schnitt $32{,}68ms$. ORB und BF nimmt im Gegensatz dazu lediglich $7{,}1ms$ in Anspruch.

Da sich, wie in Abbildung 24 gezeigt, die Genauigkeit der beiden Ansätze nur geringfügig unterscheidet, wird ORB/BF unter anderem aufgrund seiner signifikant geringeren Leistungsanforderung als Feature-Matching-Algorithmus gewählt.

Mit $7{,}1ms$ pro Bild würde pro Trainingsgeneration ein Mehraufwand von etwa 48 Minuten entstehen. Dies entspricht einer Dauer von ungefähr 33 Stunden für das Training und die Prüfung des gesamten Netzwerks bei 9 Generationen und einer Batch-size von 40.

Genauigkeit

Die Genauigkeit ist das primäre Kriterium. Sie quantifiziert, welcher Anteil aller nicht repräsentativen Ausschnitte durch die Lösung gefiltert werden kann.

Da die Überprüfung der Ausschnitte per Feature-Matching aber in zwei separate Kategorien aufgeteilt ist, müssten zunächst die einzelnen Genauigkeiten ermittelt werden.

Da die erste Kategorie, welche Fälle umfasst, wo keine Merkmale entweder im Originalbild oder im Bildausschnitt erfasst werden konnten, verhältnismäßig nur einen kleinen Teil ausmacht, wurde sich entschieden alle Ausschnitte, die darunterfallen als nicht repräsentativ einzustufen.

Sie hat somit eine Genauigkeit von 100%.

Die Genauigkeit der zweiteren Kategorie, Fälle in denen in beiden Bildern Merkmale erkannt und zugeordnet werden können, ist abhängig von dem in Kapitel 3.4.1 festgelegten Grenzwert für den Anteil von einwandfreien Zuordnungen.

Dieser wurde auf $< 2\%$ festgesetzt. Somit ergibt sich für Fälle des zweiten Falls eine Genauigkeit von $36{,}6\%$.

In Kombination wird so eine Genauigkeit von $38{,}9\%$ erreicht.

Fehlerrate

Die Fehlerrate setzt sich ebenfalls aus zwei Teilen zusammen.

- Die Fehlerrate für Fälle, in denen keine Merkmale im Ausschnitt oder Original erkannt werden
- Die Fehlerrate für Fälle, in denen zuordenbare Merkmale ausgemacht werden können

Die Fehlerrate des ersten Falles ist in Abbildung 24 dargestellt.

Da aufgrund des Anteils dieser Kategorie an der Gesamtzahl der Ausschnitte darauf verzichtet wurde eine separate Fehlerbehandlung durchzuführen, entspricht die Rate etwa 33%.

Für die zweite Kategorie entspricht die Fehlerrate dem Anteil der repräsentativen Ausschnitte, die fälschlich aussortiert werden. Diese wird aus den in Tabelle 3 bereits dargestellt Werten errechnet und beträgt 23%.

In Kombination mit der Fehlerrate der ersten Kategorie ergibt sich so eine gesamte Rate von $32{,}6\%$.

3.5 Vergleich der Ansätze

Im direkten Vergleich soll einer der beiden Ansätze ausgewählt werden, der anschließend in TReS implementiert und getestet wird.

Dazu werden beide Optionen unter mehreren Kriterien gegenübergestellt. Diese sind:

- Geschwindigkeit
- Genauigkeit
- Fehlerrate
- Komplikationsrisiko

Die Kriterien entsprechen Großteils den Gesichtspunkten, nach denen die individuellen Ansätze ausgewertet werden, werden aber noch durch den Punkt „Komplikationsrisiko“ ergänzt.

Dabei handelt es sich um eine Gegenüberstellung möglicher Probleme, die mit dem jeweiligen Verfahren entstehen könnten.

Geschwindigkeit

Im direkten Vergleich der Geschwindigkeiten benötigt die Berechnung der Entropiedifferenz etwa 4x so lange, wie das Feature Matching (siehe Abbildung 30).

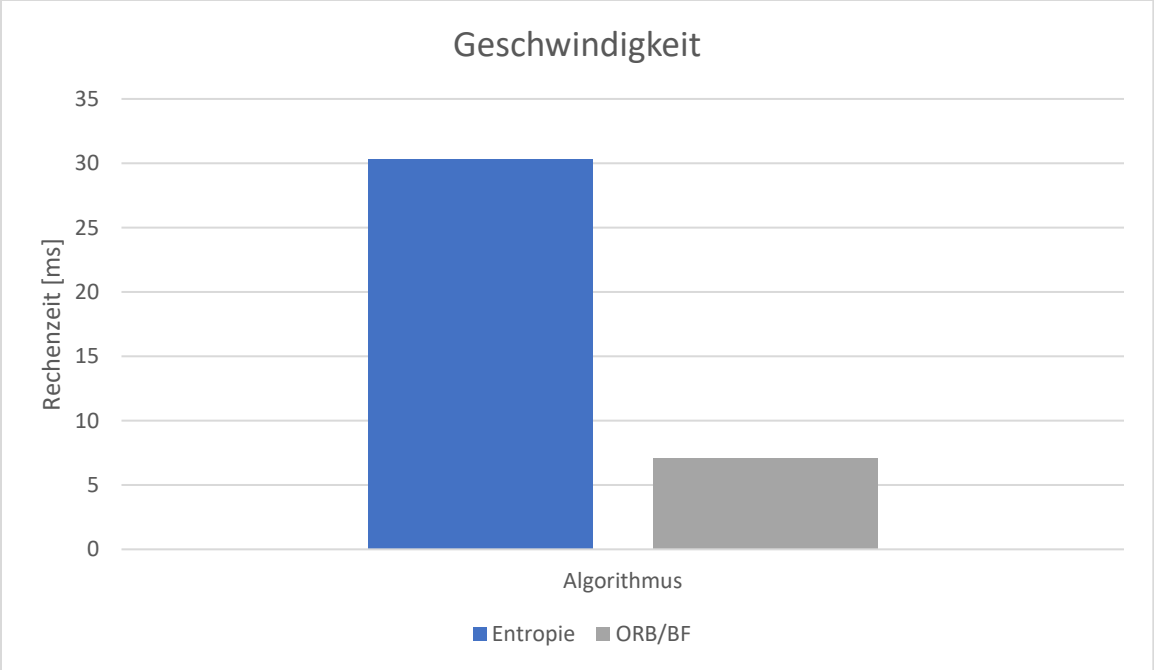


Abbildung 30 Geschwindigkeitsvergleich von Entropie und Feature Matching

Dies entspricht bei einem vollständigen Training einer Zeitdifferenz von etwa 30 Stunden.

Genauigkeit

Im Vergleich der Genauigkeiten gibt es klare Unterschiede. (siehe Abbildung 31)

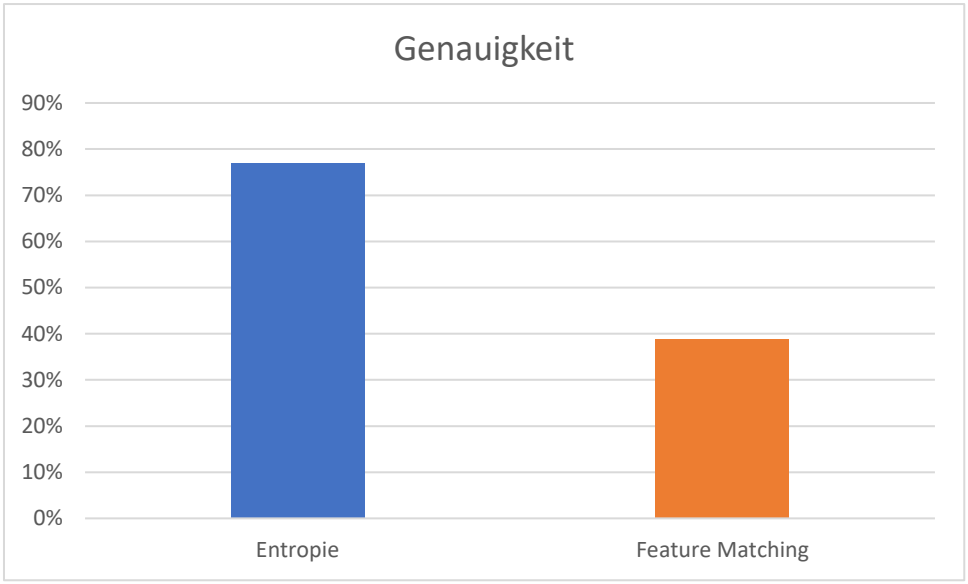


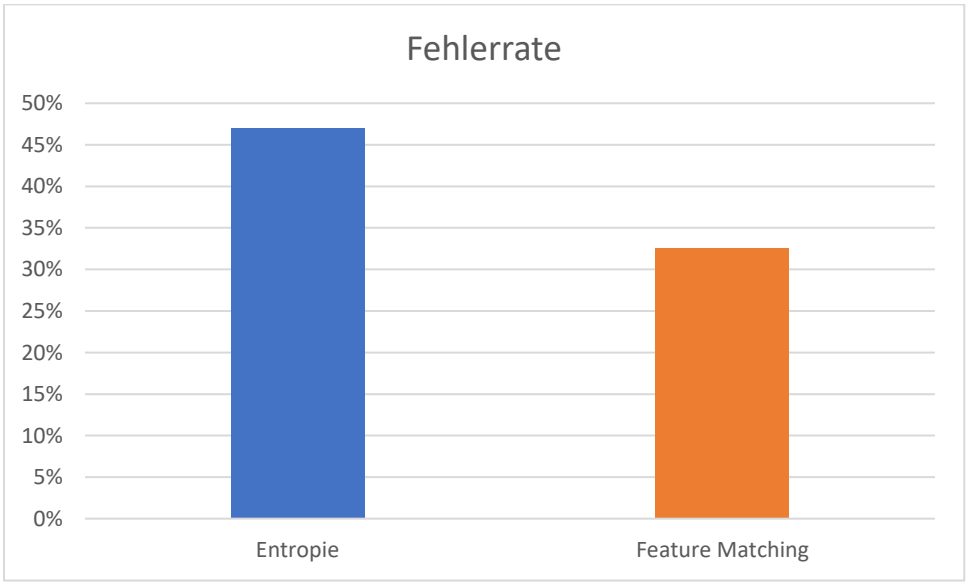
Abbildung 31 Vergleich der Genauigkeit

Die Detektion nicht repräsentativer Ausschnitte mittels der Entropiedifferenz ist hier der klare Favorit mit einer Genauigkeit von 77%.

Das Feature Matching erreicht hingegen nur eine Detektionsrate von etwa 39%.

Fehlerrate

Auch hier gibt es deutliche Unterschiede, auch wenn diese nicht so drastisch, wie bei den vorherigen Kriterien sind.



Der Entropieansatz weist eine Fehlerrate von etwa 47% auf und liegt damit oberhalb der 32,6%, die vom Feature Matching erreicht werden.

Das Feature Matching ist daher der Ansatz, bei dem weniger eigentlich repräsentative Ausschnitte fälschlicherweise herausgefiltert werden.

Tabelle 4 zeigt eine Zusammenfassung des Vergleichs.

Tabelle 4 Vergleich der beiden Ansätze

Kriterium	Entropie	Feature Matching
Geschwindigkeit	30,36ms	7,1ms
Genauigkeit	77%	39%
Fehlerrate	47%	33%

Insgesamt weist im direkten Vergleich das Filtern der Ausschnitte per Überprüfung der Entropiedifferenz mit Abstand die höchste Genauigkeit auf. Zwar ist die Rate der fälschlich entfernten Ausschnitte ebenfalls am höchsten, der Unterschied zum Feature Matching ist hier aber deutlich geringer.

Die hohe Rechenzeit wird aufgrund der großen Genauigkeitsdifferenz als vernachlässigbar erachtet.

Daher wird die Vorfilterung per Entropievergleich als Lösung ausgewählt.

Im Folgenden wird der Ansatz in TReS implementiert und überprüft.

4 Implementation

Die Implementierung der Algorithmen und Systeme erfolgt in der Programmiersprache Python. Python findet im Feld der künstlichen Intelligenz und des maschinellen Lernens weitreichend Anwendung. Dies liegt an der Flexibilität der Sprache und den vielen vorhandenen Werkzeugen, Frameworks und Bibliotheken, die die Implementierung der oft sehr komplexen und vielseitigen Systeme erleichtern. Ein weiterer Vorteil liegt darin, dass es sich bei Python um eine nutzerorientierte Allzweck-Programmiersprache handelt, was die Entwicklung deutlich beschleunigt, da es sich beim maschinellen Lernen um ein sehr experimentelles Feld handelt, bei dem oftmals viele verschiedene Prototypen entworfen, implementiert und getestet werden müssen.

Um die Komplexität zu senken, werden die entworfenen Ansätze in ein bereits vorhandenes Netzwerk zur Bildqualitätsbewertung integriert. Weitere Vorteile dieses Vorgehens sind die Möglichkeit zum Vergleich mit unabhängig erzielten Ergebnissen, gerade mit welchen, die dem neusten Stand der Entwicklung entsprechen, und die Möglichkeit zur Nutzung bereits vorhandener Werkzeuge und Infrastruktur. Die Wahl und Beschreibung dieses Netzwerks erfolgten in Kapitel 2.2.1.

Zunächst wird auf einige grundlegende Modifikationen am Netzwerk eingegangen. Anschließend wird die Implementation des Entropievergleichs und des Feature Matchings individuell betrachtet.

Da die Verarbeitung im Netzwerk sich zwar nicht bei verschiedenen Datensätzen unterscheidet, die Art und Weise, wie die Daten geladen werden jedoch unterschiedlich ist und genau dort der Ansatzpunkt liegt, wird nur exemplarisch die Implementierung für den Datensatz KonIQ [19] thematisiert. Die Vorgehensweise für andere Datensätze ist aber nahezu identisch.

4.1 Grundlagenoptimierung

Um den Entwicklungsprozess und das Testen der Konzepte zu vereinfachen, werden einige Modifikationen am Netzwerk vorgenommen.

Ein Ansatzpunkt ist die Größenänderung der Bilder während der Laufzeit des Programmes. Im Originalnetzwerk wird bei jedem geladenen Bild unter anderem die Größe auf die Maße 512x384 Pixel angepasst. Da diese Anpassung für jedes Bild, bei jeder Epoche, zweimal durchgeführt wird, ermöglicht eine Auslagerung eine erhebliche Zeitersparnis während des Trainings. Dazu wird vor Trainingsstart jedes Bild zunächst auf die gewünschte Größe gebracht und separat gespeichert. Während der Laufzeit wird dann lediglich das bereits modifizierte Bild geladen. Der restliche Programmablauf bleibt identisch.

Eine weitere Modifikation ist die Änderung des geladenen Dateiformats. Dadurch, dass die Bilder mit der im vorherigen Absatz beschriebenen Methode noch einmal separat

gespeichert werden, kommt es regelmäßig dazu, dass die geladene Version und die während der Laufzeit veränderte nicht identisch sind. Dies liegt daran, dass die Bilder im Format JPEG vorliegen. Dabei handelt es sich um ein Bildformat, bei dem eine Komprimierung der Bilder vorgenommen wird. Dadurch gehen Informationen verloren, was die Ergebnisse des Netzwerks beeinflusst.

Um dieses Problem zu beheben, wird das Netzwerk so angepasst, dass nur Bilder im Format PNG, bei dem es sich um ein verlustloses Format handelt, geladen werden können. Außerdem wird das Format der zwischengespeicherten Bilder ebenfalls angepasst. In Tests wird sichergestellt, dass die Ergebnisse nicht abweichen.

Ein weiteres Problem, welches im Rahmen der nachfolgenden Implementierungen aufgetreten ist, ist die Notwendigkeit, Zugriff auf die Koordinaten des zufällig gewählten Bildausschnittes zu haben. Da diese jedoch mit der ursprünglich gewählten Funktion *torchvision.transforms.RandomCrop()* [34] nicht zurückgegeben werden, werden die Parameter zunächst mittels der Funktion *torchvision.transforms.RandomCrop.get_params()* gewonnen und der Ausschnitt anschließend manuell erstellt.

4.2 Entropievergleich

Der Entropievergleich setzt während des Ladevorgangs der Bilddaten an. Um Rechenzeit zu sparen, werden die Entropiewerte der Gesamtbilder einmal zu Beginn des Trainings berechnet und in derselben Datei, wie die Dateipfad und Label abgelegt. Innerhalb der Trainingsschleife wird beim Laden zunächst ein zufälliger Bildindex ermittelt. Mit diesem werden der Dateipfad, das entsprechende Label und der Entropiewert des Gesamtbildes aus der CSV-Datei, in der diese abgelegt sind, geladen. Anschließend werden die Bilddaten mittels einer ausgelagerten Funktion *pil_loader()* geladen. Das eigentliche Laden erfolgt mittels der Python Imaging Library [35]. Der genaue Ablauf dieser Funktion ist in Abbildung 32 gezeigt.

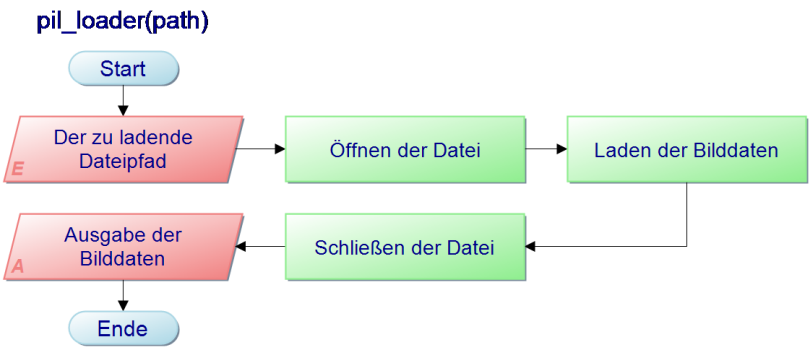


Abbildung 32 Ablaufdiagramm pil_loader()

Nachdem das Bild geladen ist, werden die vorgesehenen Transformationen durchgeführt. Diese bestehen aus dem zufälligen Spiegeln des Bildes in beiden Richtungen, dem

Ausschneiden eines kleineren Bildteils, der Konvertierung in einen Tensor und der Normalisierung dieses Tensors.

Anschließend wird die Entropie des Bildausschnittes berechnet. Dies geschieht in einer separaten Funktion und ist in Abbildung 33 dargestellt. Die vorherige Berechnung der Gesamtentropien erfolgt mittels derselben Funktion.

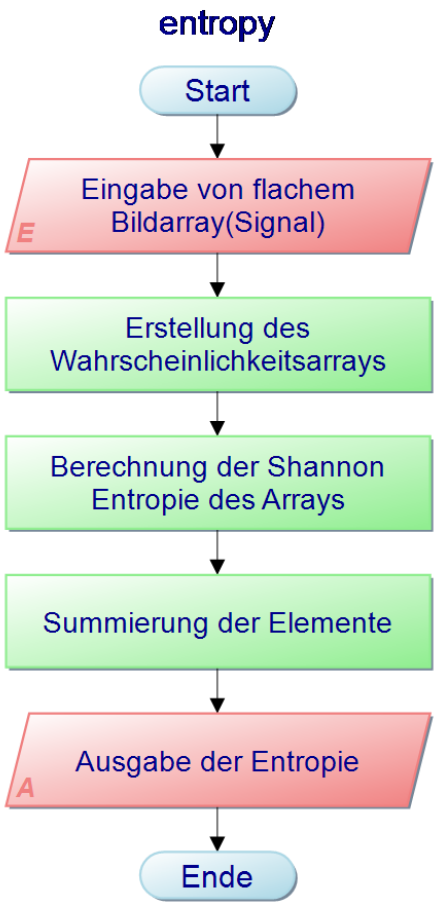


Abbildung 33 Ablaufdiagramm der Entropieberechnung

Liegen sowohl die Entropie des Gesamtbildes als auch die des Bildausschnittes vor, können diese verglichen werden. Wenn die Differenz den festgelegten Schwellwert nicht überschreitet, wird der Ausschnitt zum Training des Netzwerks verwendet. Liegt die Differenz jedoch darüber, wird zufällig ein neuer Index bestimmt und der Prozess erneut durchlaufen. Der gesamte Lade- und Vergleichsprozess ist in Abbildung 34 dargestellt.

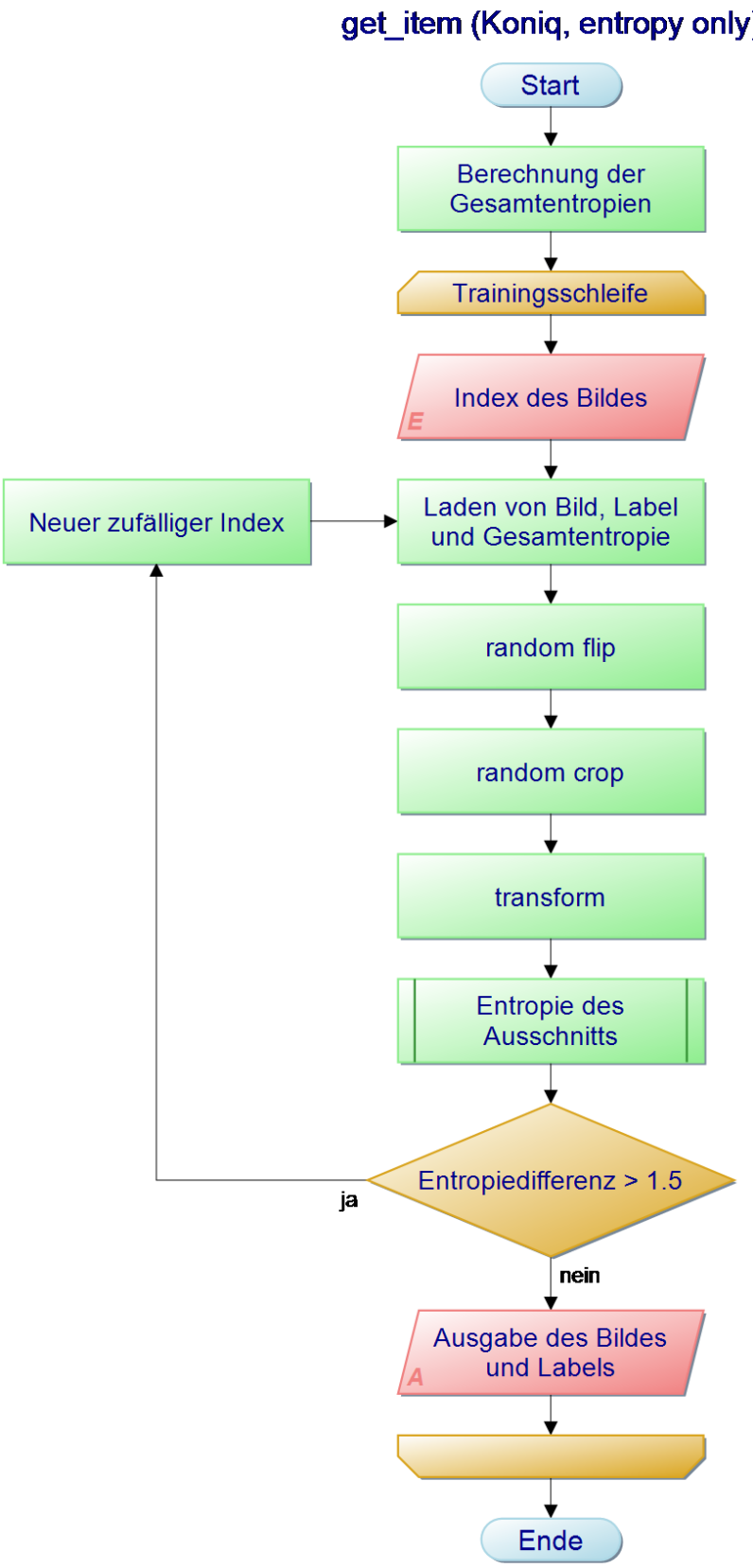


Abbildung 34 Ablaufdigramm des Entropievergleichs

4.3 Feature Matching

Das Feature Matching setzt am selben Punkt wie der Entropievergleich an. Auch hier werden das Originalbild und der Ausschnitt beim Laden der Daten verglichen. Im Gegensatz zum Entropievergleich, findet der Vergleich aber vor der Konvertierung in einen Tensor und der Normalisierung statt.

Die gewählten Feature-Extractor- und Feature-Matching-Algorithmen werden vor Beginn des Trainings initialisiert. Dabei handelt es sich um Implementationen von [36] und einen Brute-Force-Matcher aus der Python-Bibliothek OpenCV. Nachdem das Bild geladen, zufällig gespiegelt und ein Ausschnitt bestimmt wurde, werden sowohl für das Original-, als auch für das Teilbild vom Feature-Extraktor Keypoints bestimmt.

Anschließend wird geprüft, ob auch beide Bilder Keypoints enthalten. Ist dies der Fall, werden die Keypoints mittels des Brute-Force-Matching Algorithmus zugeordnet. Sollte in einem der beiden Bilder keine Keypoints vorhanden sein, wird das Bild verworfen und es wird ein neuer, zufälliger Index bestimmt.

Sind zuordenbare Merkmale vorhanden, werden diese zunächst nach ihrer Distanz geordnet. Als nächstes wird die Summe der Zuordnungen bestimmt, deren Distanz 0 entspricht. Diese werden mit der Gesamtanzahl der zugeordneten Merkmale verglichen und es wird ein Verhältnis errechnet. Liegt dieses Verhältnis unter dem in Kapitel 3.4.1 erarbeiteten Grenzwert, wird der Ausschnitt verworfen.

Der Programmablauf des gesamten Feature Matchings ist in Abbildung 35 dargestellt.

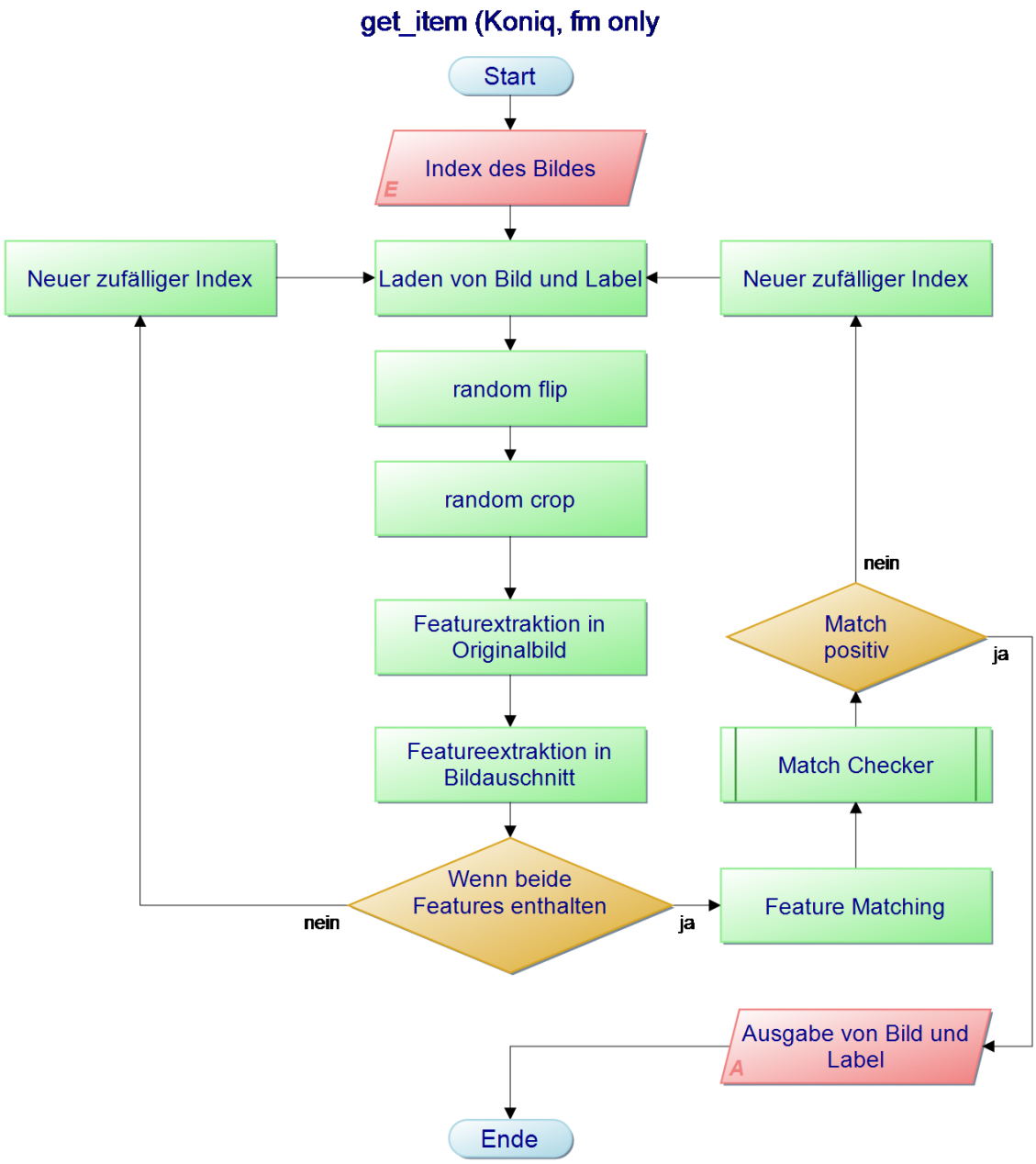


Abbildung 35 Ablaufdiagramm des Feature Matchings

5 Auswertung des Gesamtsystems

Um die Ergebnisse der Arbeit objektiv auswerten zu können, muss zunächst ein Grundwert festgelegt werden, mit dem die Ergebnisse verglichen werden können. Dazu wird das originale TReS-Netzwerk ohne eigene Modifikationen gemäß den originalen Instruktionen für den Datensatz KonIQ [19] trainiert.

Es wird darauf verzichtet weitere Datensätze zu testen, da KonIQ, wie bereits in Kapitel 3.1 dargelegt, mit Abstand der geeignetste Datensatz für Echtwelt-Anwendungen ist.

Abbildung 36 zeigt den Trainingsverlauf des Netzwerks. Dabei wird hauptsächlich der Spearman-Rank-Correlation-Coefficient (kurz SROCC) als Metrik genutzt. Ergänzend dazu wird oftmals der Pearson-Correlation-coefficient (kurz PLCC) angegeben.

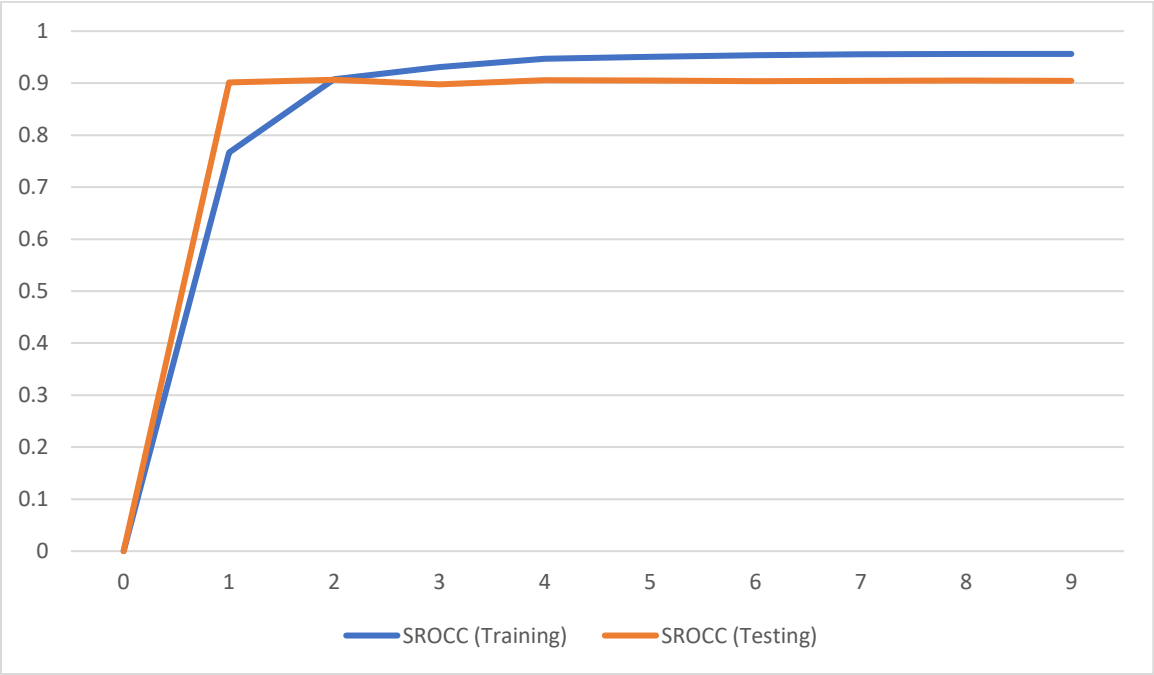


Abbildung 36 Training von TReS ohne Modifikationen

Das Netzwerk erreicht im Test mit festem Seed einen maximalen Spearman-Wert von $0{,}9065$. Die äquivalente Pearson-Korrelation entspricht $0{,}922$.

Anschließend wird das Netzwerk erneut trainiert, mit den in Kapitel 4.1 erläuterten Modifikationen. Da die Ergebnisse identisch sind, werden ein Spearman-Koeffizient von $0{,}9065$ und ein Pearson-Koeffizient von $0{,}922$ als Basiswert genommen.

Der im Paper [22] erreichte Wert von 0,915 (Spearman) und 0,928 (Pearson) konnte weder mit dem modifizierten noch dem unmodifizierten Netzwerk erzielt werden. Der Basiswert wird dennoch als repräsentativ erachtet.

Nachdem der Entropievergleich gemäß Kapitel 4.2 implementiert ist, wird das Netzwerk mit denselben Parametern trainiert, wie das Netzwerk ohne zusätzliche Filter der Eingangsdaten.

Diese Parameter sind ein Training über 9 Generationen mit einer Batchsize von 40 mit einem ResNet50 Backbone. Der Seed für den Zufallsgenerator ist identisch zum unmodifizierten Netzwerk.

Abbildung 37 zeigt den Trainingsverlauf dieses Netzwerks.

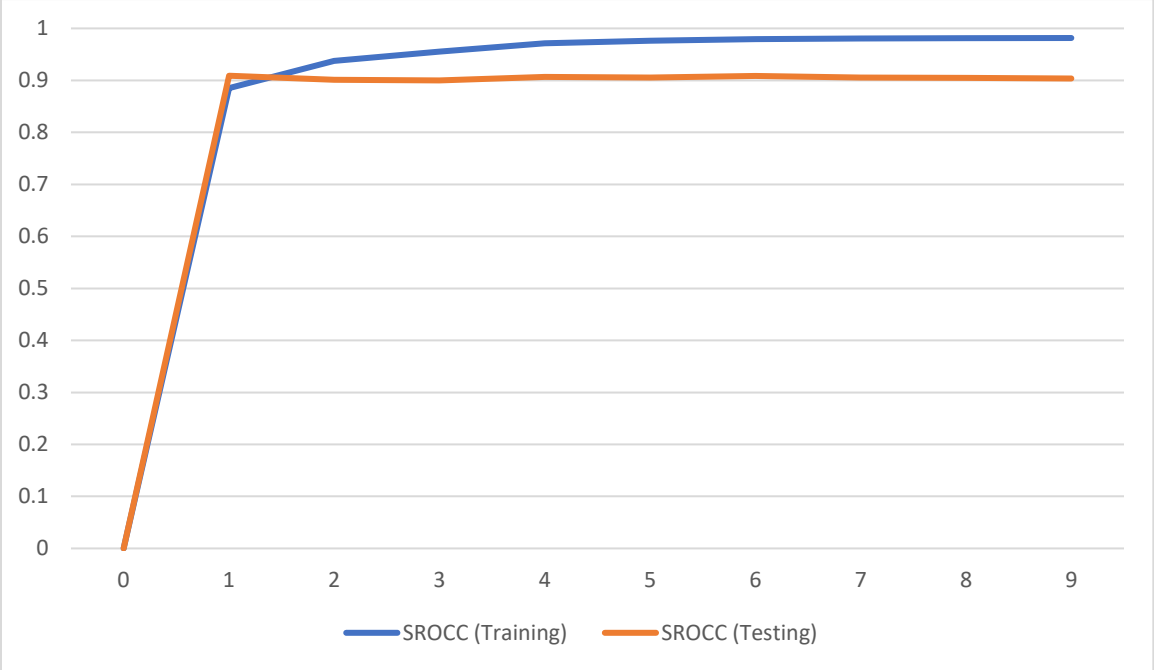


Abbildung 37 Training von TReS mit Entropiedifferenz-Filter

Das Netzwerk erreicht so einen maximalen Spearman-Koeffzienten von $0{,}9090$ und eine Pearson-Korrelation von $0{,}926$.

Im direkten Vergleich mit dem Netzwerk ohne Filter, liegen sowohl der Spearman-, als auch der Pearson-Wert, leicht oberhalb des Trainingsergebnisses des Originalnetzwerks. Diese Genauigkeitssteigerung ist jedoch zu gering, um sie einwandfrei dem Filter zuzuschreiben. Daher wird das Netzwerk zwei weitere Male, mit jeweils unterschiedlichen Seeds für den Zufallsgenerator, trainiert und es wird der Mittelwert gebildet. Dieser beträgt $0{,}9086$.

Somit kann davon ausgegangen werden, dass das Netzwerk durch die Filterung eine, wenn auch kleine, Genauigkeitssteigerung erfahren hat.

Es fällt jedoch auch auf, dass es eine erhöhte Divergenz zwischen den Ergebnissen des Trainings- und des Testdatensatzes gibt.

Das Netzwerk weist also eine höhere Tendenz zum Overfitting auf. Dieses Verhalten ist jedoch, wie in Kapitel 3.3.2 bereits gezeigt, zu erwarten. Der Grad des Overfitting fällt jedoch, im Vergleich zum Genauigkeitsgewinn, etwas höher aus, als erwartet.

6 Fazit

Das primäre Ziel dieser Arbeit war es, ein Verfahren zu entwerfen, welches in der Lage ist die Eingangsdaten von Netzwerken zur Bildverarbeitung auf Validität zu überprüfen und mögliche Falschdaten herauszufiltern.

Dieses Ziel wird als erfüllt erachtet.

Ein weiteres Ziel war die Implementation dieser Lösung in ein gängiges Netzwerk zur Bildqualitätsbewertung und das Training dieses Netzwerks.

Auch dieses Ziel wurde erfüllt.

Das sekundäre Ziel, somit eine Steigerung der Genauigkeit zu erreichen wurde teilweise erfüllt.

Das Netzwerk erreicht zwar beim Training mit ansonsten identischen Parametern eine höhere Genauigkeit, wenn die Eingangsdaten mittels des Entropievergleichs vorverarbeitet werden, es lässt sich aber nicht vollständig ausschließen, dass es sich dabei nur um eine Variation beim Training handelt, die durch eine Veränderung der Trainingsdaten entsteht.

Um eine abschließende Aussage zu treffen, wird empfohlen das Netzwerk mehrfach mit verschiedenen Seeds zu trainieren und den Mittelwert der Ergebnisse zu betrachten. Aufgrund der langen Trainingsdauer und dem begrenzten zeitlichen Rahmen der Arbeit war es nicht möglich diesen Schritt konsequent umzusetzen.

Weitere mögliche Verbesserungen der Aussagekräftigkeit sind eine Erweiterung der in den Kapiteln 3.2, 3.3 und 3.4 durchgeführten Umfragen. Da die Thematik nur einen kleineren Teil aller Bildausschnitte betrifft, ist es sehr aufwendig eine zufriedenstellende Menge an Daten zu bekommen. Eine Ausweitung hier würde die Robustheit und Zuverlässigkeit der erarbeiteten Ergebnisse deutlich steigern.

Auch eine Minimierung der Divergenz zwischen Trainings- und Testergebnis würde das festgestellte Overfitting vermindern und die Netzwerkleistung steigern.

Abschließend lässt sich sagen, dass der Inhalt dieser Arbeit eine solide Grundlage bietet, potenzielle Falschdaten vor dem Training zu erkennen, zu filtern und somit die Genauigkeit und Zuverlässigkeit von neuronalen Netzwerken zu steigern.

Dies ist nicht nur für Netzwerke zur Bildqualitätsbewertung relevant, sondern kann in nahezu allen Bereichen der Bildverarbeitung mit neuronalen Netzwerken verwendet werden.

Literaturverzeichnis

[1] U. Berkeley, „What Is Machine Learning (ML)?," 26 June 2020. [Online]. Available: https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/. [Zugriff am 24 August 2023].

[2] Wiso, „Wikipedia," 28 October 2008. [Online]. Available: https://en.wikipedia.org/wiki/Neural_network#/media/File:Neural_network_example.svg. [Zugriff am 21 August 2023].

[3] W. S. McCulloch und W. Pitts, „A LOGICAL CALCULUS OF THE IDEAS IMMANENT IN NERVOUS ACTIVITY," *Bulletin of Mathematical Biology,* pp. 99-115, 1990.

[4] „Medium," [Online]. Available: https://miro.medium.com/v2/resize:fit:4800/format:webp/0*oyFmAciLmqLVxeB0.jpg. [Zugriff am 26 Dezember 2023].

[5] Y. LeCun, L. Buttou, Y. Bengio und P. Haffner, „Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE, vol. 86,* pp. 2278-2324, November 1998.

[6] C. Cortes und V. Vapnik, „Support-Vector Networks," *Machine Learning,* pp. 273-297, 1995.

[7] J. Huang, J. Chai und S. Cho, „Deep learning in finance and banking: A literature review and classification," *Frontier of Business Research in China 14,* 2020.

[8] K. Kourou, P. E. Konstantinos, C. Papaloukas, P. Sakaloglou, T. Exarchos und D. I. Fotiadis, „Applied machine learning in cancer research: A systematic review for patient diagnosis, classification and prognosis," *Comput Struct Biotechnol J.,* pp. 5546-5555, 6 October 2021.

[9] Y. Huang und Y. Chen, „Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies," 2020. [Online]. Available: https://arxiv.org/abs/2006.06091. [Zugriff am 31 Januar 2023].

[10] M. Nath und S. Sagnika, „Capabilities of Chatbots and Its Performance Enhancements in Machine Learning," *Machine Learning and Information Processing. Advances in Intelligent Systems and Computing, vol 1101,* 24 März 2020.

[11] H. R. Sheikh, M. F. Sabir und A. C. Bovik, „A Statistical Evaluation of Recent Full Reference," *IEEE Transactions on Image Processing, Vol. 15, No. 11,* pp. 3440-3451, November 2006.

[12] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li und S. Hu, „Traffic-Sign Detection and Classification in the Wild," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* Juni 2016.

[13] S. Sonawane und A. Deshpande, „Image Quality Assessment Techniques: An Overview," *International Journal of Engineering Research & Technology, Vol. 3, Issue 4,* pp. 2013-2017, April 2014.

[14] V. Kamble und K. Bhurchandi, „No-reference image quality assessment algorithms: A survey," *Optik 126,* pp. 1090-1097, Mai 2015.

[15] H. Sheikh, Z. Wang, L. Cormack und A. Bovik, „LIVE Image Quality Assessment Database Release 2,“ [Online]. Available: http://live.ece.utexas.edu/research/quality. [Zugriff am 26 September 2023].

[16] N. Ponomarenko, L. Jin, O. Ieremeiev, V. Lukin, K. Egiazarian, J. Astola, B. Vozel, K. Chehdi, M. Carli, F. Battisti und C.-C. J. Kuo, „Image database TID2013: Peculiarities, results and perspectives,“ *Signal Processing: Image Communication,* Nr. 30, pp. 55-77, 2015.

[17] E. Larson und D. Chandler, „Most apparent distortion: full-reference image quality assessment and the role of strategy,“ *Journal of Electronic Imaging,* Nr. 19(1), 2010.

[18] Z. Ying, H. Niu, P. Gupta, D. Mahajan, D. Ghadiyaram und A. Bovik, „From Patches to Pictures (PaQ-2-PiQ): Mapping the Perceptual Space of Picture Quality,“ 20 Dezember 2019. [Online]. Available: https://arxiv.org/abs/1912.10088.

[19] V. Hosu, H. Lin, T. Sziranyi und D. Saupe, „KonIQ-10k: An ecologically valid database for deep learning of blind image quality assessment,“ *IEEE TRANSACTIONS ON IMAGE PROCESSING ,* Bd. 29, pp. 4041-4056, 2020.

[20] Y. Fang, H. Zhu, Y. Zeng, K. Ma und Z. Wang, „Perceptual Quality Assessment of Smartphone Photography,“ 2020.

[21] S. Su, Q. Yan, Y. Zhu, C. Zhang, X. Ge, J. Sun und Y. Zhang, „Blindly Assess Image Quality in the Wild Guided by A Self-Adaptive Hyper Network,“ 2020.

[22] S. A. Golestaneh, S. Dadsetan, K. M. Kitani, C. M. University und U. o. Pittsburgh, „No-Reference Image Quality Assessment via Transformers, Relative Ranking, and Self-Consistency,“ 5 Januar 2022. [Online]. Available: https://arxiv.org/pdf/2108.06858.pdf. [Zugriff am 29 November 2022].

[23] K. He, X. Zhang, S. Ren und J. Sun, „Deep Residual Learning for Image Recognition,“ 10 Dezember 2015. [Online]. Available: https://arxiv.org/pdf/1512.03385.pdf. [Zugriff am 20 November 2023].

[24] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov und S. Zagoruyko, „End-to-End Object Detection with Transformers,“ 2020.

[25] „TReS,“ [Online]. Available: https://user-images.githubusercontent.com/12434910/137831770-dd5d17da-fe83-431e-ac86-bebbe2810820.png. [Zugriff am 29 September 2023].

[26] H. Lin, V. Hosu und D. Saupe, „KADID-10k: A Large-scale Artificially Distorted IQA Database,“ in *2019 Eleventh International Conference on Quality of Multimedia Experience*, 2019.

[27] D. Ghadiyaram und A. C. Bovik, „Massive Online Crowdsourced Study of Subjective and Objective Picture Quality,“ *IEEE Transactions on Image Processing,* Bd. 25, 2015.

[28] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth und L.-J. Li, „FCC100M: The new data in multimedia research,“ *Communications of the ACM,* Bd. 59, Nr. 64-73, 2016.

[29] C. Sun, A. Shrivastava, S. Singh und A. Gupta, „Revisiting Unreasonable Effectiveness of Data in Deep Learning Era,“ 4 August 2017. [Online]. Available: https://arxiv.org/abs/1707.02968. [Zugriff am 29 November 2022].

[30] S. Yang, T. Wu, S. Shi, S. Lao, Y. Gong, M. Cao, J. Wang und Y. Yang, „MANIQA: Multi-dimension Attention Network for No-Reference Image Quality Assessment," 21 April 2022. [Online]. Available: https://arxiv.org/pdf/2204.08958.pdf. [Zugriff am 29 November 2022].

[31] C. E. Shannon, „A mathematical theory of communication," *Bell System Techn. J,* p. 379–423; 623–656, 1948.

[32] OpenCV, „Feature2D Class Reference," [Online]. Available: https://docs.opencv.org/3.4/d0/d13/classcv_1_1Feature2D.html. [Zugriff am 06 Dezember 2022].

[33] F. K. Noble, „Comparison of OpenCV's feature detectors and feature matchers," *2016 23rd International Conference on Mechatronics and Machine Vision in Practice (M2VIP),* pp. 1-6, 2016.

[34] „PyTorch Documentation," [Online]. Available: https://pytorch.org/vision/main/generated/torchvision.transforms.RandomCrop.html. [Zugriff am 10 September 2023].

[35] J. A. Clark, „Pillow (PIL Fork) Documentation," 2015. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf. [Zugriff am 11 Oktober 2023].

[36] E. Rublee, V. Rabaud, K. Konolige und G. Bradski, „ORB: An efficient alternative to SIFT or SURF," *2011 International Conference on Computer Vision,* pp. 2564-2571, 2011.

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Bachelorarbeit selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Die vorgelegte Arbeit hat weder in der gegenwärtigen noch in einer anderen Fassung schon einem anderen Fachbereich der Hochschule Ruhr West oder einer anderen wissenschaftlichen Hochschule vorgelegen.

Duisburg, 26.12.2023

Ort, Datum

Unterschrift